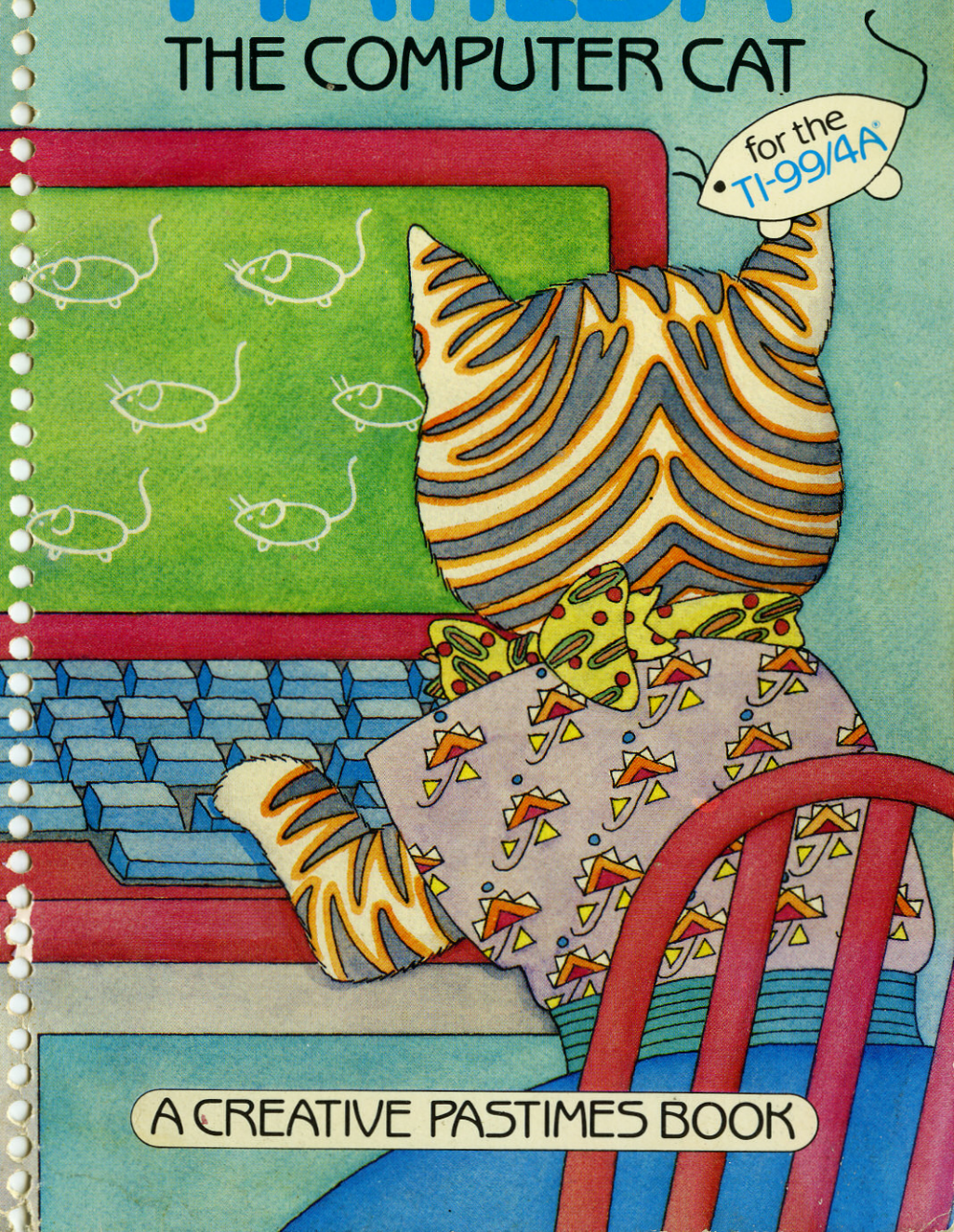


CLAIRE BAILEY PASSANTINO

MATILDA

THE COMPUTER CAT



A CREATIVE PASTIMES BOOK

Matilda the Computer Cat for the TI-99/4A[®] Computer

Claire Bailey Passantino

Text Illustrations by Nancy Gurganus



**A Reston Computer Group Book
Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia**

A CREATIVE PASTIMES BOOK

Library of Congress Cataloging in Publication Data

Passantino, Claire Bailey.

Matilda, the Computer Cat, for the TI-99/4A computer.

(A Creative pastimes book)

"A Reston Computer Group book."

1. TI-99/4A (Computer)—Programming. 2. Basic
(Computer program language) 3. Computer games. I. Title.
II. Series.

QA76.8.T133P38

1984

001.64'25

83-26964

ISBN 0-8359-4324-0

TI-99/4A® is a registered trademark of Texas Instruments.

Cover illustration by Bethann Thornburgh

Cover design by Carol Conway

© 1984 by Claire Bailey Passantino

All rights reserved. No part of this book may be reproduced
in any way, or by any means, without permission in writing
from the author and the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Other Itty Bitty Bytes Books:

**Itty Bitty Bytes of Space
School Days**

**These titles are also available for
the ATARI® Computer and
the Commodore 64® Computer**



Contents

Foreword, vii
A Note to Parents and Teachers, ix
Meet Matilda, 2–3
The Dating Game, 4–5
Our Song, 6–7
Does He Love Me?, 8–9
Heart Throbs, 10–11
A New Rug!, 12–13
Fleas!, 14–15
Nine Lives, 16–17
Name That Baby, 18–19
The Cat’s Pajamas, 20–21
Hey Diddle Diddle, 22–23
Lost Mittens, 24–25
Do You Know Your Cats?, 26–27
Cost of Living, 28–29
A Merry Chase, 30–31
In the Kitty, 32–33
Fat Cat, 34–35
Final Exam, 36–37
What a Party!, 38–39
A New Career, 40–41
Appendix, 43



Foreword

I hope you enjoy Matilda as much as I've enjoyed writing about her. She has worked her way into my heart as I've worked her into my computer. Maybe the same thing will happen to you.

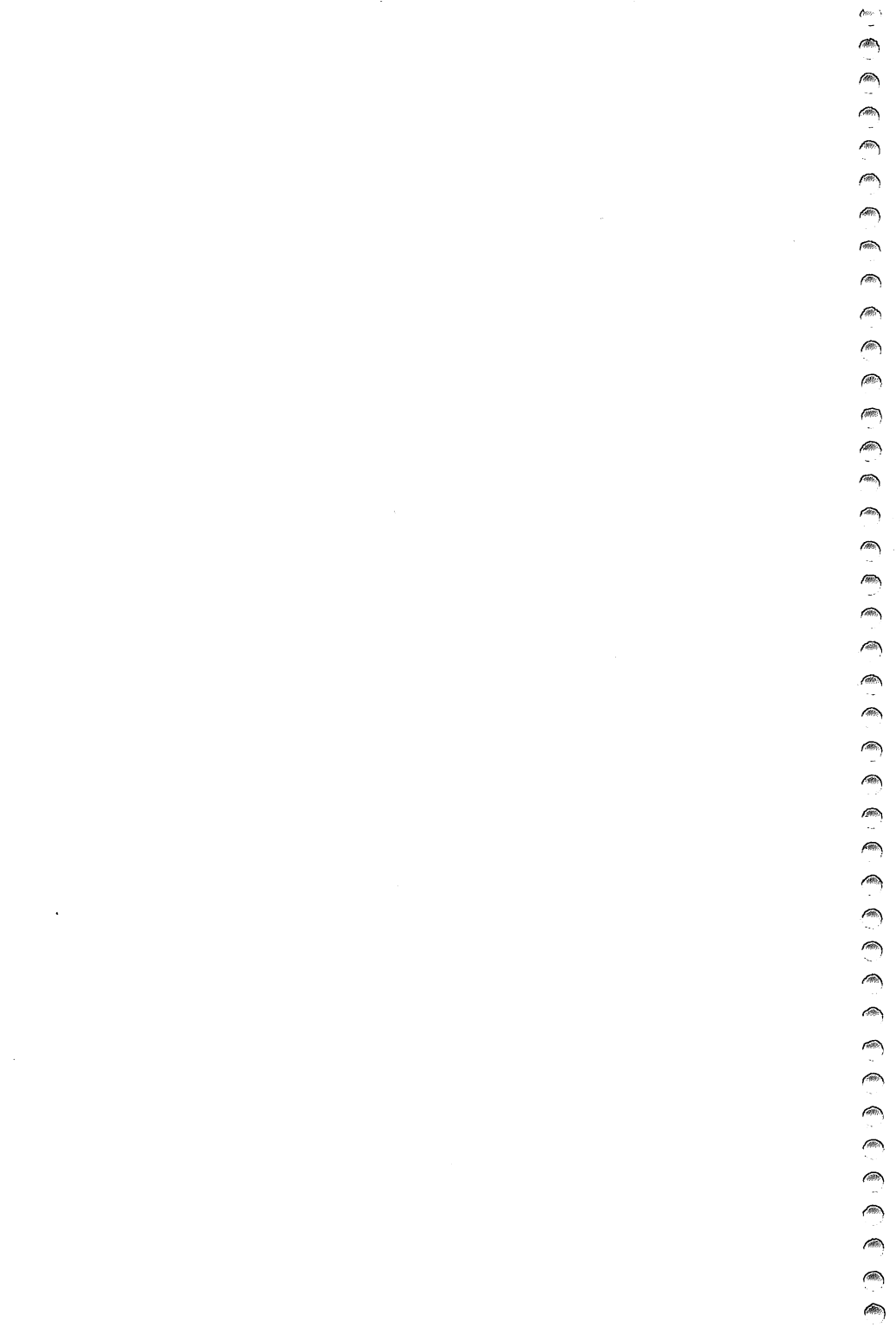
This is the second of the Itty Bitty Bytes books for the TI-99/4A® Computer. Each book in the series is designed to bring you twenty fun programs and, at the same time, teach you something about BASIC. Typing in the programs is hard work. You must be careful to number the lines exactly, spell the words correctly, and put in all the right punctuation. But you will feel extra good when your program runs smoothly!

Always read the explanations next to each program. Sometimes there are additions or changes that you can make to improve the program. Once you see what the program is all about, feel free to experiment with it—and by all means combine programs that work well together. The more you can do to make a program reflect your own special personality, the more you will enjoy it. Let me know if you find some nifty changes!

The Itty Bitty Bytes books have grown out of the teaching experiences I've had with my computer students. Special thanks go to all my "COMPU-KIDS"—and to my own kids, too—who keep bringing me one good idea after another. They are my severest critics, but my most outstanding assets.

Happy computing!

Claire Bailey Passantino



A Note to Parents and Teachers

You bought the computer. You read the manuals. You did the demo programs. You and/or your children may even have taken some computer classes. But now the computer is just sitting there. Everyone was so enthusiastic. *What happened?*

The novelty of a new computer will take you just so far. Beyond that point, a continuing interest in this incredible tool is directly related to its usefulness. "Not useful" equates to "not used." So the problem becomes, what can computers do that children would find useful? What kinds of things would encourage children to expend the energy needed to create their own computer programs?

Each Itty Bitty Bytes book in the Creative Pastimes series is packed with computer activities appealing to young programmers. Simple games, contests, races, pictures, designs, songs, riddles, charts, tests, and more—all are designed to be fun while reinforcing beginning computer concepts and skills.

Besides providing fun, there are fringe benefits to having children write their own programs. In working with children, I have found that computer programming encourages them to:

- Think creatively.
- Use logical thinking skills.
- Attend to details.
- Take small steps to achieve a goal.
- Personalize programs.
- Develop pride and self-esteem.
- Appreciate packaged software.

To help children enjoy doing their own programs, here are some suggested DOs and DON'Ts:

- DO encourage children to type in their own programs. With younger children, bargain: "You type this line and I'll type the next one." (Save the long lines for yourself!)
- DON'T criticize typing expertise. Speed and correct fingering are typing skills that are minimally related to computer programming.
- DO allow children to make mistakes.
- DO help them find the errors they've made. (This is called "debugging" the program.)
- DON'T worry when there is an error message. This means that a mistake has been made. Check the program for "bugs."
- DO encourage children to read and understand the program explanations.

-
- DON'T, however, force the issue. Some people learn by reading. Others learn by doing. As skills are repeated over and over in different contexts, children may just "catch on."
 - DO be aware of some common pitfalls. Remember to:
 1. Use line numbers.
 2. Press ENTER after you type in a line.
 3. Give great attention to spelling and punctuation. Quotation marks, commas, semicolons, colons, and even spaces are very important.
 4. Save your program before you turn off the computer.
 - DO help children save their programs on tape or disk so they can use them again and again. If you have a printer, use it to make "hard" copies of each program. People like to see themselves in print.
 - DON'T type the "FCTN" and the plus (+) keys simultaneously. This combination will erase the program from memory and you will have to start over.
 - DO remember that typing the "FCTN" and the "4" simultaneously will break the program and get you out of an infinite loop.
 - DO type in and run all programs with the ALPHA LOCK on (that is, in the *down* position). The only exceptions are the joystick programs, which are run with the ALPHA LOCK off.
 - DO praise children for a job well done. And enjoy the programs that they've created.
 - DO modify and use programs that you yourself find useful.

It is my sincere hope that the Itty Bitty Bytes books will help you and your children establish a healthy working relationship with your computer. Take that computer off the shelf! And let me know how things are working out!

**Matilda the Computer Cat
for the TI-99/4A[®] Computer**

Main ideas: Creating a design using the keyboard characters
Practice in using the keyboard

- To use a character located on the front of a key, hold down the FCTN key as you type.
- To use the uppermost character on a two-character key, hold down the SHIFT key as you type.

5 Clears the screen.

10–160 Makes the cat. Special notes are listed below:

10 \ is FCTN–Z. ! is SHIFT–1. " is FCTN–P.

30–40 * is SHIFT–8.

50 The bottom of Matilda's smile is FCTN–U.

60 Her collar is a row of = signs.

70 Her back is FCTN–U.

130 This row uses ! and - (which is SHIFT–/).

160 Her paws are SHIFT–9, comma, and SHIFT–0.

250 This is a dummy line; it prevents the DONE message from appearing. Leave it out and see what happens. To get out of the infinite (never-ending) loop, press FCTN–4.

Lines 170 to 230 allow you to criticize the drawing:

170 Skips a line after the cat is drawn.

180–190 Pauses while the computer counts to itself to 1000. (Change the 1000 for a longer or shorter pause.)

200 Prints the question.

210 Waits for your response. The dollar sign (\$) is called a "string." The string allows you to respond with words or alphabetic information. Without the string, the computer expects you to respond with a number.

220 Skips a line.

230 Prints a message.

Making designs or pictures by using keyboard characters can be a lot of fun. Why don't you try your own?



Meet Matilda

Type in the program and you can meet Matilda.

```
5 CALL CLEAR
10 PRINT " / \ / \ !!"
20 PRINT " / \ / \ !!"
30 PRINT " ! * * ! !!"
40 PRINT " \ \ * / / !!"
50 PRINT " \ \ --- / / !!"
60 PRINT " \ ===== / !!"
70 PRINT " / \ ----- / /"
80 PRINT " / !"
90 PRINT " !"
100 PRINT " !"
110 PRINT " !"
120 PRINT " !"
130 PRINT " ! !-! !--! !-! !"
140 PRINT " ! ! ! ! ! ! ! !"
150 PRINT " ! ! ! ! ! ! ! !"
160 PRINT " ( , ) ( , ) ( , ) ( , )"
250 GOTO 250
```

If you don't like my drawing, tell me so:

```
170 PRINT
180 FOR P=1 TO 1000
190 NEXT P
200 PRINT "DO YOU LIKE THE WAY I LOOK?"
210 INPUT A$
220 PRINT
230 PRINT "NOW DON'T BE CATTY!"
```

Main idea: A vertical race up the screen.

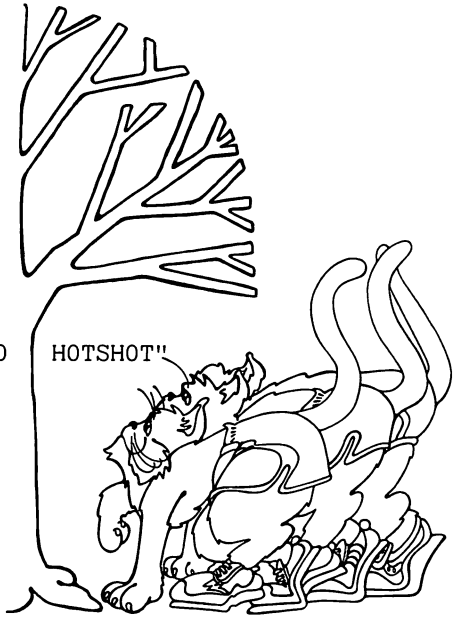
During this program, Hairy (H), Macho (M), and Hotshot (HS) race to the top of their "trees." Character 97 in set 9 is used for the "trees." Character 104 in set 10 is used to draw a square for Hairy. Character 112 in set 11 is used to draw a square for Macho. Character 120 in set 12 is used to draw a square for Hotshot. Character 40 in set 2 is used to draw a square under each cat as he climbs up the tree. The color of set 2 is the same as the background color. In effect then, the "old" cat is erased as the new character is drawn.

- 5–10 Clears the screen and makes it light blue (color 6).
- 15–20 Makes the characters in set 2 light blue on light blue (solid blue squares) and makes the characters in set 9 black on black.
- 30–70 Draws 3 "trees" at columns 8, 16, and 24.
 - 80 Prints the names of the characters under their trees.
- 90–91 Hairy will start at row 22. Hairy's square (in set 10) will be dark red (color 7).
- 100–101 Macho will start at row 22. Macho will be dark yellow (color 11).
- 110–111 Hotshot will start at row 22 and will be magenta (color 14).
 - 120 Sets the random function. Leave this out and run the program a few times to see what happens.
 - 130 Picks a number from 1 to 3.
- 140–155 If a 1 is picked in line 130, this is a vote for Hairy, and you are sent to line 200. If a 2 is picked, this is a vote for Macho, and you are sent to line 300. If a 3 is picked, you go through lines 160 to 195.
 - 160 Subtracts 1 from Hotshot (HS).
 - 170 Draws the new HS somewhere in column 23 (next to the tree).
 - 180 Draws a square the same color as the background right under the new HS (on top of the old HS).
 - 190 If HS is at the top of the tree (at row 1), you go to line 400.
 - 195 If HS is not at the top of the tree, you go back to line 130 and the computer picks a number from 1 to 3 all over again.
- 200–250 Subtracts 1 from Hairy (H) and redraws a box for H on the first tree (in column 7). Blanks out the old box for H. If H has reached row 1, the computer skips to line 400. Otherwise, it goes back to line 130.
- 300–350 Subtracts 1 from Macho (M) and redraws a box for M on the second tree (in column 15). Blanks out the old box for M. If M has reached row 1, the computer skips to line 400. Otherwise, it goes back to line 130.
- 400–410 Asks you who won and waits for you to type in the winner's name.
 - 420 Prints a repeating message.
 - 430 Goes back to line 420. This sets the program into an infinite loop. Type FCTN-4 to break out of the program.

For fun, change the colors of the cats (lines 91, 101, and 111). Or change their names (line 80).

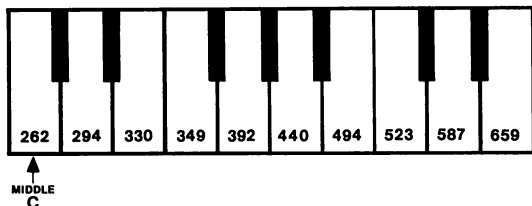
The Dating Game

```
5 CALL CLEAR
10 CALL SCREEN(6)
15 CALL COLOR(2,6,6)
20 CALL COLOR(9,2,2)
30 FOR C=8 TO 24 STEP 8
40 FOR R=22 TO 1 STEP -1
50 CALL HCHAR(R,C,97)
60 NEXT R
70 NEXT C
80 PRINT "   HAIRY   MACHO   HOTSHOT"
90 H=22
91 CALL COLOR(10,7,7)
100 M=22
101 CALL COLOR(11,11,11)
110 HS=22
111 CALL COLOR(12,14,14)
120 RANDOMIZE
130 X=INT(3*RND+1)
140 IF X=1 THEN 200
150 IF X=2 THEN 300
155 REM HOTSHOT CLIMBS UP
160 HS=HS-1
170 CALL HCHAR(HS,23,120)
180 CALL HCHAR(HS+1,23,40)
190 IF HS=1 THEN 400
195 GOTO 130
200 REM HAIRY CLIMBS UP
210 H=H-1
220 CALL HCHAR(H,7,104)
230 CALL HCHAR(H+1,7,40)
240 IF H=1 THEN 400
250 GOTO 130
300 REM MACHO CLIMBS UP
310 M=M-1
320 CALL HCHAR(M,15,112)
330 CALL HCHAR(M+1,15,40)
340 IF M=1 THEN 400
350 GOTO 130
400 PRINT "WHO WON?";
410 INPUT W$
420 PRINT "   ";W$;"   WINS A DATE!"
430 GOTO 420
```



Three cool cats are vying for Matilda's attention. She tells them that she will date the best tree climber. Who will win?

Main idea: Using READ . . . DATA to play a song



The numeric equivalents of some of the notes are shown in the little diagram above. A more complete chart is on page 53. To play a song on the computer, data lines are used to tell the computer which notes to play and how many beats to hold each note. The first 3 notes in this song (line 100) are note 330 for 3 beats, note 294 for 3 beats, and note 262 for 6 beats.

- 10 Clears the screen.
- 20 Prints a message.
- 30 The colons advance the message up the screen.
- 40 Tells the computer to search for a data line and read the first 2 numbers. The first number will be N (the note), and the second number will be B (the number of beats).
- 50 The number “-1” will be used as the “flag” data. Flag data is used in a program as a signal to the computer. In this case, when the computer reads a -1 as the note, the program skips to line 200.
- 60 Plays the note. Each beat will be 150 milliseconds long. To make the song slower, increase this number. To play it faster, decrease the number. The loudest volume is 1.
- 70 Goes back to line 40 to read the next note and the number of beats it gets.
- 100-130 The data itself.
- 190 The flag data.
- 200 Prints a final message.
- 210 Moves the message up a few lines.
- 220 Freezes the message on the screen. (Press FCTN-4 to break the program.)

Here is the rest of the song:

- 140-170 More data lines to finish the song.

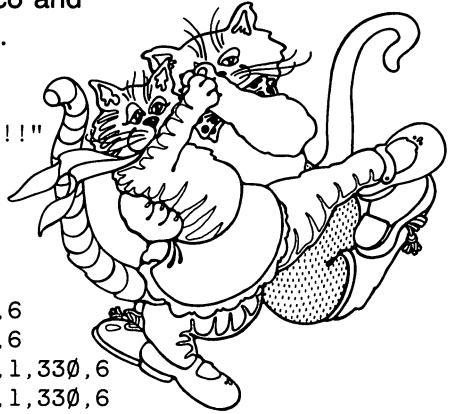
For a little color, try the following:

- 35 Picks a number from 1 to 16.
- 36 Clears the screen with the color picked.
- 70 Goes back to line 35 so that the screen changes color with each note.

Our Song

Matilda decides that she likes dancers better than tree climbers. She meets a really good dancer named Tom. Every night they go to the neighborhood disco and dance to their favorite song.

```
10 CALL CLEAR
20 PRINT "          OUR SONG!!"
30 PRINT: : : : : : : :
40 READ N,B
50 IF N=-1 THEN 200
60 CALL SOUND(B*150,N,1)
70 GOTO 40
100 DATA 330,3,294,3,262,6
110 DATA 330,3,294,3,262,6
120 DATA 392,3,349,2,349,1,330,6
130 DATA 392,3,349,2,349,1,330,6
190 DATA -1,-1
200 PRINT "ALL THE CATS LOVE THIS ONE!"
210 PRINT : : : : :
220 GOTO 220
```



Want to hear the rest of the song?

```
140 DATA 392,1,523,2,523,1,494,1,440,1,494,1,
523,2,392,1,392,2
150 DATA 392,1,523,2,523,1,494,1,440,1,494,1,
523,2,392,1,392,2
160 DATA 392,1,523,2,523,1,494,1,440,1,494,1,
523,2,392,1,392,2
170 DATA 349,1,330,3,294,3,262,6
```

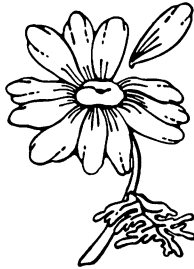
How about a flash dance?

```
35 C=INT(16*RND+1)
36 CALL SCREEN(C)
70 GOTO 35
```

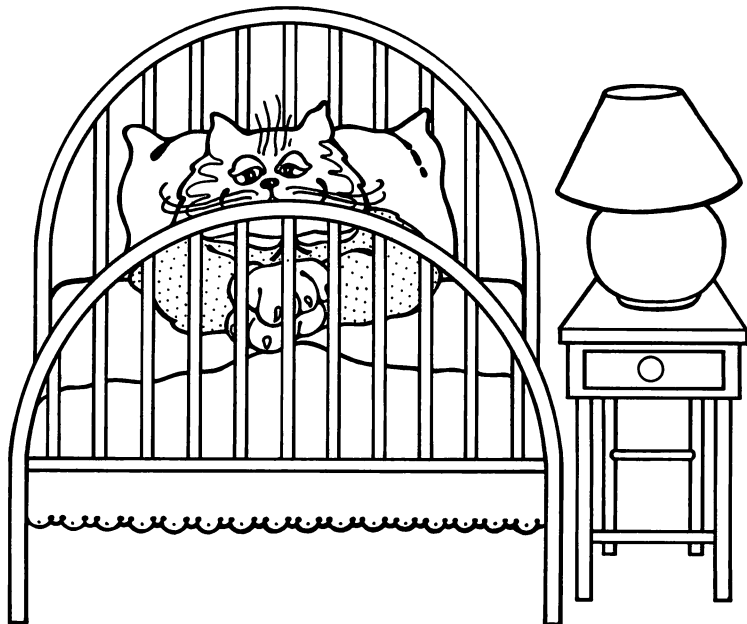
Main ideas: Randomizing
Incrementing a variable

- 10 LM (loves me) is set to 0.
- 20 LMN (loves me not) is set to 0.
- 30 Randomizes the selection of numbers.
- 40–90 Everything in this loop will be done 365 times:
 - 40 Q (the big question) will be asked 365 times.
 - 50 The computer picks a number, 0 or 1.
 - 60 If it picked a 0, skips to line 75.
 - 70 By default if it picked a 1, 1 is added to the LMN total.
 - 71 Skips to line 80.
 - 75 When a 0 is picked, the LM total is increased by 1.
 - 80 Prints the LM total.
 - 85 Prints the LMN total.
 - 86 Skips a line.
 - 90 Finishes a loop.
 - 95 Before the final message is printed, skips a few lines.
- 100 If the LM total is more than half of 365, goes to line 150.
- 110 Prints a message.
- 120 Ends.
- 150 Prints a message.

What a silly way to make a decision! If you want to make the decision in less than a year, change the number in line 40 and the halfway mark in line 100.



Does He Love Me?



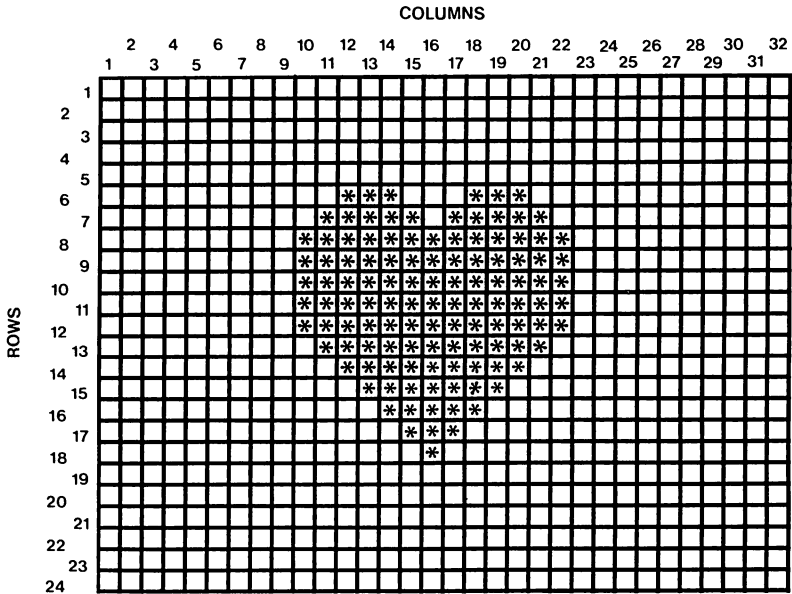
Every night Matilda asks herself, "Does he love me?" After one year, this is what she decides:

```
10 LET LM=0
20 LET LMN=0
30 RANDOMIZE
40 FOR Q=1 TO 365
50 X=INT(2*RND)
60 IF X=0 THEN 75
70 LMN=LMN+1
71 GOTO 80
75 LM=LM+1
80 PRINT "LOVES ME",LM
85 PRINT "LOVES ME NOT",LMN
86 PRINT
90 NEXT Q
95 PRINT:::
100 IF LM>=183 THEN 150
110 PRINT "HE HATES ME! RATS!!"
120 END
150 PRINT "HE LOVES ME! EXCELLENT!!"
```

Main ideas: Using the Graphics Chart
Special effects with random colors

- 10 Clears the screen.
- 20 Makes the screen white.
- 30 To make the heart, we are going to use character 42 in set 2. This line makes set 2 include red characters on a white background.
- 40–160 Draws the heart. One horizontal line is produced at a time by filling in the required information: CALL HCHAR (row number, column number, character 42, how many). Lines 100 to 120 produce lines of asterisks (*) at rows 12, 11, 10, 9, and 8. Each row starts in the 10th column and extends through 13 boxes.
- 170–200 The “throbbing” section.
 - 180 Picks a random number from 1 to 16.
 - 190 The characters in set 2 (which includes the character 42 that we are using) are all changed to the color bearing the number selected in line 180.
 - 200 Goes back to line 180 to pick another color.

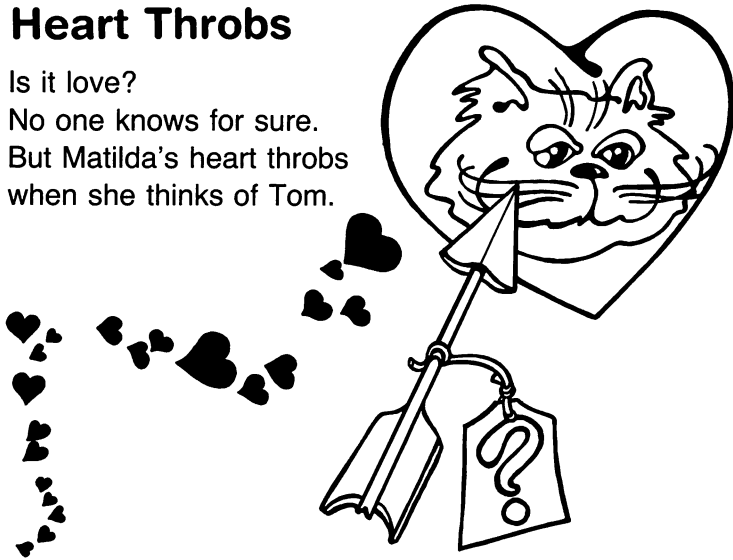
As you type in the program, you may like to refer to this diagram.



Heart Throbs

Is it love?

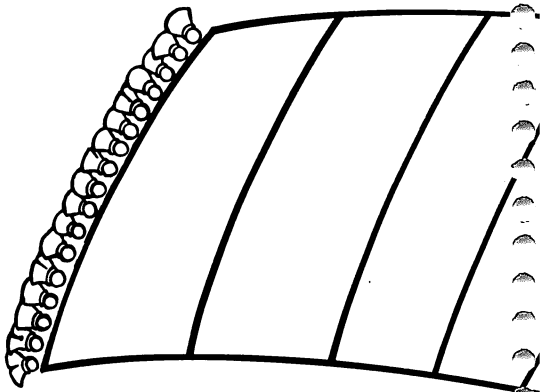
No one knows for sure.
But Matilda's heart throbs
when she thinks of Tom.



```
10 CALL CLEAR
20 CALL SCREEN(16)
30 CALL COLOR(2,9,16)
40 CALL HCHAR(18,16,42,1)
50 CALL HCHAR(17,15,42,3)
60 CALL HCHAR(16,14,42,5)
70 CALL HCHAR(15,13,42,7)
80 CALL HCHAR(14,12,42,9)
90 CALL HCHAR(13,11,42,11)
100 FOR R=12 TO 8 STEP -1
110 CALL HCHAR(R,10,42,13)
120 NEXT R
130 CALL HCHAR(7,11,42,5)
140 CALL HCHAR(7,17,42,5)
150 CALL HCHAR(6,12,42,3)
160 CALL HCHAR(6,18,42,3)
170 REM HEART THROBS
180 C=INT(16*RND+1)
190 CALL COLOR(2,C,16)
200 GOTO 180
```

Main idea: Using HCHAR and VCHAR to fill in an area with a pattern

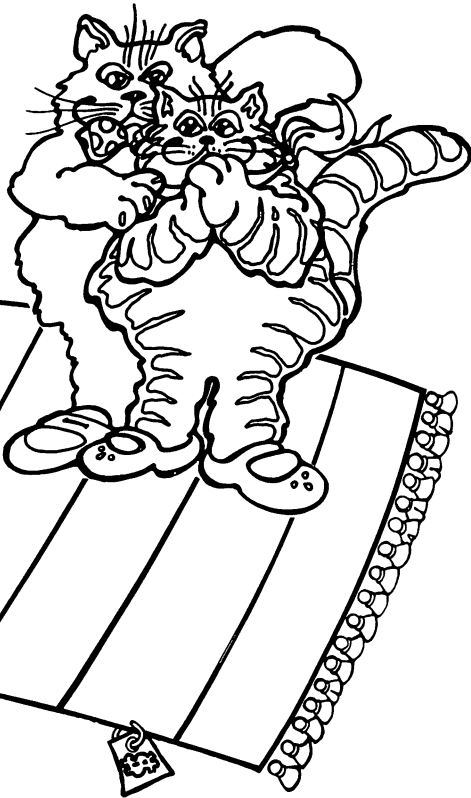
- 10 Clears the screen.
- 15–50 Draws the rug:
 - 20 We are going to use the characters in set 4 to make the rug. We will start out with dark blue (color 5) on light red (color 10). You can change this combination if you like.
 - 30 We are going to fill in rows 5 through 50.
 - 40 Starting at the row we're up to (R) in column 6, repeats character 60 for 20 times. Set 4 contains characters 56 to 63. You may change character 60 to another number for a different effect.
 - 50 Completes the loop.
- 60–90 Makes the fringe using character 95, which is similar to an underline:
 - 70 Character 95 is in set 8. We are making set 8 blue (color 5) on a green (color 4) background. Since the screen is green, only the blue "fringe" will show up. Change the 5 to another number for a different color of fringe.
 - 80 Makes a column of fringe in column 5.
 - 90 Makes a column of fringe in column 26.
- 100 An infinite loop—a dummy line that keeps going to itself. Press FCTN-4 to break out of the program. (Omit line 100 to find out why you need a dummy line.)



A New Rug!

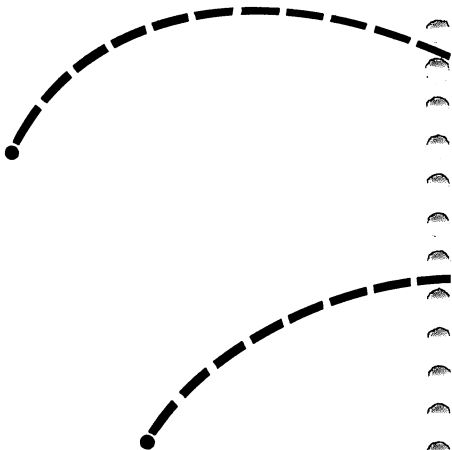
Tom and Matilda went to a big flea market one weekend. What a beautiful rug Matilda bought. And so cheap, too!

```
10 CALL CLEAR
15 REM RUG
20 CALL COLOR(4,5,10)
30 FOR R=5 TO 20
40 CALL HCHAR(R,6,60,20)
50 NEXT R
60 REM FRINGE
70 CALL COLOR(8,5,4)
80 CALL VCHAR(4,5,95,17)
90 CALL VCHAR(4,26,95,17)
100 GOTO 100
```



Main idea: Guessing game

- 10 Clears the screen.
- 20–40 Prints the directions.
- 50 Sets the random numbers.
- 60 Picks a number from 0 to 199.
- 70 Allows you to input a guess.
- 80 If your guess was exactly right, the computer goes to line 200.
- 90 If your guess was too low, it skips to line 120.
- 100 If your guess was too high, then you will end up at this line since lines 80 and 90 do not apply. This line prints "FEWER!"
- 110 Sends you back to line 70 to guess again.
- 120 Prints "MORE!"
- 130 Sends you back to line 70 to guess again.
- 200–230 As a reward for guessing correctly, you get to see a march of the fleas:
- 210 Sets up the loop.
- 220 Prints an apostrophe. (Leave off the semicolon for a change.)
- 230 Completes the loop and goes back to line 210 until the required number of loops are completed.



Fleas!

Unfortunately, the rug had fleas. Poor Matilda! Guess how many fleas she counted just today!

```
10 CALL CLEAR
20 PRINT "POOR MATILDA HAS FLEAS!"
30 PRINT "GUESS HOW MANY SHE HAS."
40 PRINT "HINT:LESS THAN 200!"
50 RANDOMIZE
60 F=INT(200*RND)
70 INPUT G
80 IF G=F THEN 200
90 IF G<F THEN 120
100 PRINT "FEWER!"
110 GOTO 70
120 PRINT "MORE!"
130 GOTO 70
200 PRINT "HERE THEY COME!"
210 FOR N=1 TO F
220 PRINT " ";
230 NEXT N
```



Main ideas: Initializing, incrementing, and decrementing variables

- 10 Clears the screen.
- 20 L stands for the number of lives Tom has. At the beginning of the game, Tom has 9 lives.
- 30 Sets the random numbers.
- 40 T stands for Tom's location. The computer picks a number from 1 to 5. Each number represents one of Tom's girlfriends.
- 50–60 Prints the directions.
- 70 Allows you to input your guess. G stands for the number you pick.
- 75 Clears the screen.
- 80 If you did not guess the correct girlfriend Tom is visiting, you are sent back to line 40 and the computer picks a girlfriend for Tom to visit the next day. (It can pick the same number again.)
- 90 If you make it to this point, it means you have guessed Tom's location accurately.
- 100 Tom loses a life. The number of lives (L) is decreased by 1.
- 110 If Tom loses all of his lives ($L=0$), the computer skips to line 200.
- 120 Otherwise, it tells you how many lives Tom has left.
- 130 Goes back to line 40 so you can continue guessing where Tom goes next. (Poor Tom never learns and just keeps losing those lives!)
- 200 When Tom has lost all his lives, the screen is cleared.
- 220 The message is printed and moves up the screen. (The colons move it up the screen.)

If you want to see how many visits Tom gets away with, add the lines indicated.

- 25 Sets the number of visits (V) to 0. (At the beginning of the game, Tom has visited no one.)
- 45 Every time the computer picks a new number in line 40, 1 is added to the number of visits.
- 230 The tally of the number of visits is printed at the end of the game.

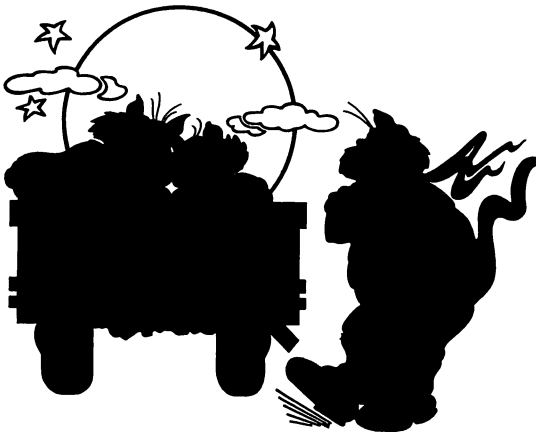
Nine Lives

Oh no! Tom is a creep. He has *five* other girlfriends besides Matilda. Every day he visits one of them. If Matilda finds out, she will just kill him. You may play this game until Tom loses all of his nine lives.

```
10 CALL CLEAR
20 L=9
30 RANDOMIZE
40 T=INT(5*RND+1)
50 PRINT "OK,MATILDA!"
60 PRINT "WHERE IS TOM TODAY?(1-5)"
70 INPUT G
75 CALL CLEAR
80 IF G<>T THEN 40
90 PRINT "YOU'RE DEAD,TOM!"
100 L=L-1
110 IF L=0 THEN 200
120 PRINT L;" LIVES TO GO!":
130 GOTO 40
200 CALL CLEAR
210 CALL SCREEN(6)
220 PRINT "      BYE BYE TOM!!":
```

Add these lines to find out how long he got away with it.

```
25 V=0
45 V=V+1
230 PRINT "BUT YOU GOT AWAY WITH IT ";V;" TIMES."
```



Main ideas: A race

Distinguishing between numeric and string variables

Flashing colors

- 10–30 All 3 names will start the “race” at box 3. This is called *initializing the variables*.
- 40 Clears the screen.
- 50–100 You are asked to input the 3 names. Notice the semicolons at the end of lines 50, 70, and 90. Leave them out and see what happens. Also notice the dollar sign (\$) after the variable names in lines 60, 80, and 100. The \$ sign makes the variables into “string” variables. The computer will accept alphabetic information for string variables, but not for numeric variables.
- 110 Clears the screen.
- 120–140 Each name is printed and advanced 6 rows up the screen. This positions the names for the beginning of the race.
- 150 Sets the random seed.
- 160 Picks a number from 1 to 3.
- 170 If a 1 is picked in line 160, the computer goes to line 200. If a 2 is picked, it goes to line 300. If a 3 is picked, it goes to line 400.
- 200–230 Tells what happens if the computer picked a 1 in line 160:
- 200 Character 42 is drawn at row 7 and column N1 (whatever number N1 is up to).
- 210 N1 is incremented by 1.
- 220 Each row extends to column 32. So if any name reaches box 33, the row is completed and you are sent to line 500.
- 230 If the row is incomplete, you are sent back to line 160 where the computer can give another “vote” for one of the names.
- 300–330 Tells what happens if the computer picked a 2 in line 160.
- 400–430 Tells what happens if the computer picked a 3 in line 160.
- 500 Ends the program.

Character 42, the asterisk (*), is in set 2. You may decide to experiment with the colors of the asterisks. For example, you may try:

```
CALL COLOR (2,5,15) or  
CALL COLOR (2,14,5)
```

The first combination will give you dark blue asterisks on a gray background; the second will give you magenta asterisks on a dark blue background. See the Color Chart on page 47 and find color combinations that you like.

- 500–530 A flashy ending:
- 500 Makes 30 flashes by counting from 1 to 30.
- 510 Picks a number from 1 to 16.
- 520 Gives the screen the color picked in line 510.
- 530 Sends you back to line 500 for the required number of loops.

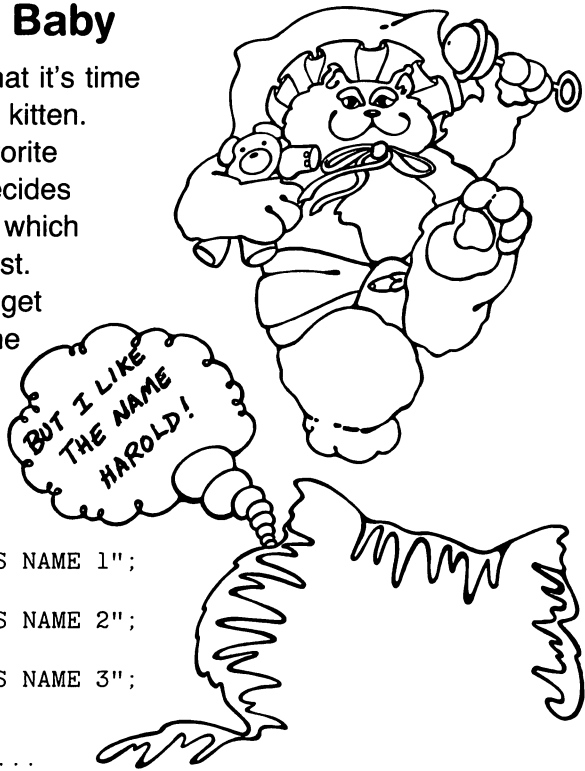
Name That Baby

Matilda decides that it's time to name her baby kitten.

She has three favorite names, so she decides to ask her friends which name they like best.

The first name to get 32 votes will be the winner. Maybe.

```
10 N1=3
20 N2=3
30 N3=3
40 CALL CLEAR
50 PRINT "WHAT IS NAME 1";
60 INPUT N1$
70 PRINT "WHAT IS NAME 2";
80 INPUT N2$
90 PRINT "WHAT IS NAME 3";
100 INPUT N3$
110 CALL CLEAR
120 PRINT N1$:::
130 PRINT N2$:::
140 PRINT N3$:::
150 RANDOMIZE
160 X=INT(3*RND+1)
170 ON X GOTO 200,300,400
200 CALL HCHAR(7,N1,42)
210 N1=N1+1
220 IF N1=33 THEN 500
230 GOTO 160
300 CALL HCHAR(13,N2,42)
310 N2=N2+1
320 IF N2=33 THEN 500
330 GOTO 160
400 CALL HCHAR(19,N3,42)
410 N3=N3+1
420 IF N3=33 THEN 500
430 GOTO 160
500 END
```



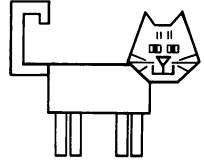
Experiment with color:

```
165 CALL COLOR(2,14,5)
```

For a flashy finish:

```
500 FOR FL=1 TO 30
510 C=INT(16*RND+1)
520 CALL SCREEN(C)
530 NEXT FL
```

Main ideas: Nested FOR . . . NEXT loops
Using the CALL KEY statement



- 10 Clears the screen.
- 20 Makes the screen white (color 16).
- 30–90 Draws the body of the pajamas:
 - 40 We are going to use character 42, which is in set 2. To start with, we are going to have blue squares (color 5). (Leave this line out and see what happens.)
 - 50 R stands for the rows. Try adding “STEP 2” to this line. STEP 2 makes the computer count by twos.
 - 60 C stands for the columns. Try adding “STEP 2” to this line.
 - 70 Sets the squares, one at a time.
 - 80 The end of the C loop. This returns you to line 60 until all the columns are counted.
 - 90 The end of the R loop. This returns you to line 50 so you can go on to the next row.

Notice how the above loops are nested. If the loops were crossed, the program would not work. If you switch rows 80 and 90, for example, the loops would be improperly nested. This arrangement of loops will *not* work:

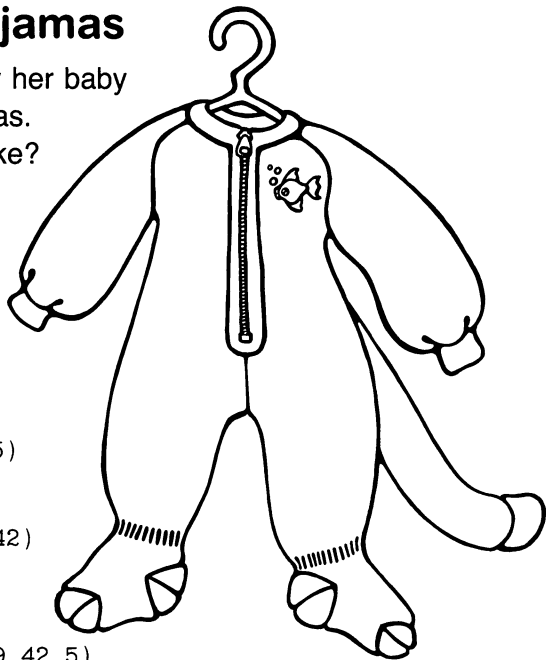
```
50 FOR R=6 TO 14 STEP 2
60 FOR C=8 TO 24 STEP 2
70 CALL HCHAR(R, C, 42)
80 NEXT R
90 NEXT C
```

- 100–140 Creates 4 vertical lines for the sleeves. Each sleeve begins at row 15 and goes down 5 boxes.
- 150–160 Prints the directions.
- 170–210 This loop allows you to see the colors available. When the shade (SH) of the color is 1 (transparent) or 16 (the same color as the background), the pajamas “disappear.”
 - 170 Picks a number for the shade of the color.
 - 180 Sets the color of the characters in set 2.
 - 190 Sets up the “PRESS ANY KEY” situation.
 - 200 “STATUS=0” means that no key has been pressed, and you keep going back to line 190. When you finally do press a key, you will go on to line 210.
 - 210 Sends you back to line 170 so the computer can pick a new color for the pajamas.

(If you hold down the key, the pajamas will flash as the colors are changing rapidly!)

The Cat's Pajamas

Matilda wants to buy her baby
a new pair of pajamas.
Which pair do you like?



```
10 CALL CLEAR
20 CALL SCREEN(16)
30 REM BODY
40 CALL COLOR(2,5,5)
50 FOR R=6 TO 14
60 FOR C=8 TO 24
70 CALL HCHAR(R,C,42)
80 NEXT C
90 NEXT R
100 REM SLEEVES
110 CALL VCHAR(15,9,42,5)
120 CALL VCHAR(15,13,42,5)
130 CALL VCHAR(15,19,42,5)
140 CALL VCHAR(15,23,42,5)
150 PRINT "    PRESS ANY KEY"
160 PRINT "    TO CHANGE COLORS!"
170 SH=INT(16*RND+1)
180 CALL COLOR(2,SH,SH)
190 CALL KEY(0,N,STATUS)
200 IF STATUS=0 THEN 190
210 GOTO 170
```

Press any key to see a different color. Read the directions to get stripes or checks!

Main idea: Inserting variables into a text

This program has two sections:

10–160 In this section, the computer asks you to input information.

200–300 In this section, it clears the screen and puts the information you gave it into a text. The text here is the poem “Hey Diddle Diddle.”

Note the use of spaces and semicolons. The semicolon acts as a formatting direction to the computer. It tells the computer: The first thing to be printed after the semicolon will appear *right next* to the last thing printed before the semicolon. For example, look at line 260:

```
260 PRINT "OVER THE ";SK$
```

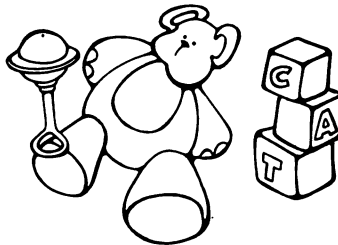
↑
space

If you put in the word MOON for SK\$, your printout would read:

```
OVER THE MOON
```

If you leave out the space, it would read:

```
OVER THEM OON
```



Hey Diddle Diddle

Matilda is teaching nursery rhymes to Harold. Sometimes she forgets the words. Maybe you can help her out!

```
10 REM GET INPUTS
20 CALL CLEAR
30 PRINT "PLEASE GIVE ME:"
40 PRINT "A MUSICAL INSTRUMENT";
50 INPUT I$
60 PRINT "AN ANIMAL";
70 INPUT AN$
80 PRINT "SOMETHING IN THE SKY";
90 INPUT SK$
100 PRINT "ANOTHER ANIMAL";
110 INPUT AN2$
120 PRINT "2 THINGS AT THE DINNER TABLE"
130 PRINT " 1.";
140 INPUT TH$
150 PRINT " 2.";
160 INPUT TH2$
200 REM PRINT POEM
210 CALL CLEAR
220 CALL SCREEN(14)
230 PRINT "HEY DIDDLE DIDDLE"
240 PRINT "THE CAT AND THE ";I$
250 PRINT "THE ";AN$;" JUMPED"
260 PRINT "OVER THE ";SK$
270 PRINT "THE LITTLE ";AN2$;" LAUGHED"
280 PRINT "TO SEE SUCH SPORT"
290 PRINT "AND THE ";TH$;" RAN AWAY"
300 PRINT "WITH THE ";TH2$::: :
```

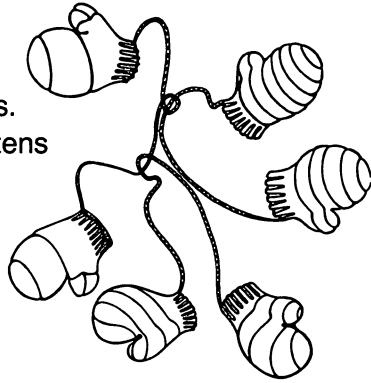


Main idea: Using the arrow keys to draw on the screen

- 10 Sets the random numbers.
 - 20 This is the beginning of a very long loop that ends at line 420. The loop will be executed 3 times—once for each pair of lost mittens.
 - 30–40 The beginning row and column are set at 1,1 (the upper left corner).
 - 50 Clears the screen.
 - 60 Makes the screen white (color 16).
 - 70 Picks a number from 0 to 13 and adds 2 to that number. The end result is a number from 2 to 15. This eliminates color 1 (which is transparent) and color 16 (which would match the background color). If colors 1 or 16 were used, any drawing would be invisible.
 - 80 We are going to draw with character 42, which is in set 2. This line sets the color of set 2 to the color chosen in line 70.
 - 90 Prints the question.
 - 100–110 Picks the row (M) and column (N) where the mittens are hidden.
 - 120 The beginning of the search: Your starting point is drawn. As R and C are changed during the program, you keep coming back to this line so your current location can be drawn.
 - 130 If the row where you are (R) matches the row where the mittens are (M), you are sent to line 400 to check the column location.
 - 140 CALL KEY is used to see if you have pressed a key.
 - 150 If you have not pressed a key, the computer goes back to line 140 until you *do* press a key.
 - 160–190 If any of the keys below are pressed, you are sent to an appropriate line where 1 is added or subtracted to the number of rows and columns. The program then goes back to line 120 to plot your new location. CHR\$(69) is E, which is where the ↑ arrow is located.
CHR\$(88) is X, which is where the ↓ arrow is located.
CHR\$(68) is D, which is where the → arrow is located.
CHR\$(83) is S, which is where the ← arrow is located.
 - 400 If the row where you are matches the row where the mittens are, the computer checks to see if the columns match. If there is no match, it goes back to line 140.
 - 410 If you have matching rows and columns, this message is printed and moves up the screen. (The colons move it up.)
 - 420 Goes back to the beginning of the loop until all 3 pairs of mittens are found.
 - 430–440 Occurs when all 3 pairs of mittens are found.
- Adding the extra lines gives you these results:
- 210–290 Prevents error messages, which occur when the computer tries to draw at a location that is not on the screen. Columns 1 to 32 and rows 1 to 24 are the only columns and rows on the screen.
 - 450–480 A flashy ending.

Lost Mittens

Harold and his two friends are always losing their mittens. Please help the three little kittens and *find those mittens!*



```

10 RANDOMIZE
20 FOR T=1 TO 3
30 R=1
40 C=1
50 CALL CLEAR
60 CALL SCREEN(16)
70 HUE=INT(14*RND+2)
80 CALL COLOR(2,HUE,HUE)
90 PRINT "WHERE ARE MY MITTENS?"
100 M=INT(24*RND+1)
110 N=INT(32*RND+1)
120 CALL HCHAR(R,C,42)
130 IF R=M THEN 400
140 CALL KEY(0,KEY,STATUS)
150 IF STATUS=0 THEN 140
160 IF KEY=69 THEN 200
170 IF KEY=88 THEN 250
180 IF KEY=68 THEN 300
190 IF KEY=83 THEN 350
200 R=R-1
220 GOTO 120
250 R=R+1
270 GOTO 120
300 C=C+1
320 GOTO 120
350 C=C-1
370 GOTO 120
400 IF C<>N THEN 140
410 PRINT "YAY!YOU FOUND
THEM!!":;
420 NEXT T
430 PRINT " THREE CHEERS FOR YOU!"
440 PRINT " MEW MEW MEW!!":;

```

To keep your drawing on the screen:

```

210 IF R<1 THEN 230
230 R=1
240 GOTO 120
260 IF R>24 THEN 280
280 R=24
290 GOTO 120
310 IF C>32 THEN 330
330 C=32
340 GOTO 120
360 IF C<1 THEN 380
380 C=1
390 GOTO 120

```

For a flashy ending:

```

450 FOR FL=1 TO 25
460 B=INT(16*RND+1)
470 CALL SCREEN(B)
480 NEXT FL

```

Main idea: Using READ . . . DATA to design a test

- 10 Clears the screen.
- 20 Prints the question and moves it up the screen.
- 30 Reads the "cat" word and its description from a data line. (The first data to be read is in line 200.)
- 40 If the word "END" has been read, the computer skips to line 400.
- 50 Otherwise, the description is printed.
- 60 Allows you to guess the "cat" word. Spelling counts. If you put in even one extra space, your word will not match the computer's desired "cat" word.
- 65 If you did not match the computer's word, you are sent to line 90 where you receive the correct answer.
- 70 If you did match the computer's word, you go to line 71. (This line is unnecessary, but it helps you see the logic.)
- 71 GOSUB means that the program goes to line 100 (for some flashing) until it gets to the word "RETURN" in line 150. Then the program comes right back here and continues where it left off.
- 80 Goes back to the beginning to give you the next word.
- 90–95 Occurs when you give an incorrect response:
 - 90 Skips a line.
 - 91 Gives you the correct "cat" word, C\$.
- 92–93 Pauses while the computer counts to itself to 500.
- 95 Goes back to line 10 for the next word.
- 100–150 Occurs when you give a correct response:
 - 110 This loop will be executed 25 times.
 - 120 Picks a number from 1 to 16. We called it B (for background).
 - 121 If B is 1 (transparent) or 2 (black), it is unacceptable since the words cannot be read if the letters match the background color.
 - 130 Changes the screen color to that chosen in line 120.
 - 140 Completes the loop.
 - 150 Returns to the main program at line 80.
- 200–270 The data for the program. Data lines may be added anywhere before line 300. If you can think of more "cat" words, be sure to include them here!
- 300 The flag data. When the computer reads the word "END," line 40 tells it to skip to line 400 to end the program.
- 400–430 Ends the program.

Do You Know Your Cats?

Matilda wants her baby to grow up proud to be a cat. Every day as he gets older, Matilda teaches him the words that every cat should know. To check his vocabulary, sometimes she gives him this little test.

```
10 CALL CLEAR
20 PRINT "DO YOU KNOW YOUR CATS?": : : : :
30 READ C$,D$
40 IF C$="END" THEN 400
50 PRINT D$
60 INPUT G$
65 IF G$<>C$ THEN 90
70 IF G$=C$ THEN 71
71 GOSUB 100
80 GOTO 10
90 PRINT
91 PRINT "WRONG. IT WAS ";C$
92 FOR P=1 TO 500
93 NEXT P
95 GOTO 10
100 REM REWARD FOR RIGHT ANSWER
110 FOR F=1 TO 25
120 B=INT(16*RND+1)
121 IF B=1 THEN 120
122 IF B=2 THEN 120
130 CALL SCREEN(B)
140 NEXT F
150 RETURN
200 DATA CATERPILLAR,A BABY BUTTERFLY
210 DATA CATNIP,A CAT'S FAVORITE PLANT
220 DATA CATCHER,A BASEBALL PLAYER
230 DATA CATALOG,WHERE CATS BUY THEIR CLOTHES
240 DATA CATTLE,A HERD OF CATS
250 DATA CATSUP,KETCHUP FOR KITTIES
260 DATA CATASTROPHE,A TRIP TO THE VET
270 DATA CATCHOO,WHEN CATS SNEEZE
300 DATA END,END
400 REM ENDING
410 PRINT "    CAT'S ALL FOLKS!": :
420 PRINT "CAN YOU THINK OF ANY OTHERS?"
430 END
```

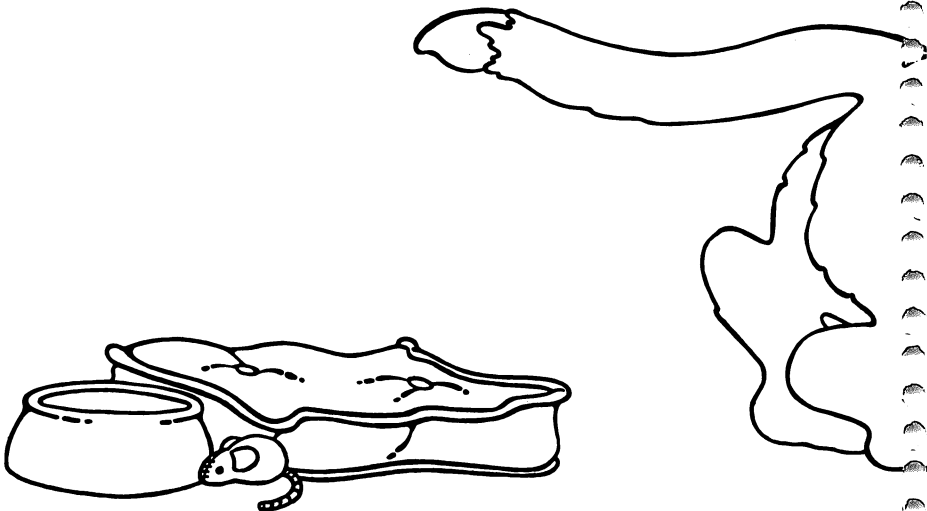


Main ideas: Making a chart
Using conversion formulas

- 10 Clears the screen
- 20–50 Allows you to input a number for the cost of a can of cat food. This number should be entered as a decimal. For example, enter .37 for 37 cents, *not* 37¢ or \$.37 or 37. You also are allowed to input the number of cans used in a day. C stands for the cost of a can and N stands for the number of cans.
- 60 Clears the screen again.
- 70–80 Prints the title of the chart and underlines the title. Notice the spaces.
- 90 Skips a line.
- 100 D stands for the daily cost, which is computed by multiplying the cost (C) by the number of cans (N). Notice that the computer uses a star (*) for multiplying.
- 110–140 Prints the information for the chart. Notice the spaces. They are used to make your chart look neat. The comma is used to send the information into 2 columns, called *print zones*. The semicolon is used to print the number *right next to* the dollar sign. Remember from line 100 that D stands for the daily cost. It stands to reason, then, that other costs can be figured as follows:

$$\begin{aligned}\text{Weekly cost} &= 7 * D \\ \text{Monthly cost} &= 30 * D \\ \text{Yearly cost} &= 365 * D\end{aligned}$$

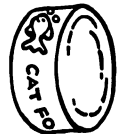
- 150 Moves the chart up the screen, centering it.



Cost of Living

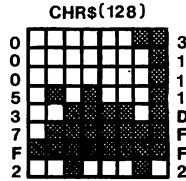
Harold has developed quite an appetite. Matilda is trying to figure out how much money she spends on cat food. You can do this, too. Just type in this simple program!

```
10 CALL CLEAR
20 PRINT "HOW MUCH DOES ONE CAN COST?"
30 INPUT C
40 PRINT "HOW MANY CANS DO WE USE IN ONE DAY?";
50 INPUT N
60 CALL CLEAR
70 PRINT "      COST OF CAT FOOD"
80 PRINT "      _____"
90 PRINT
100 D=C*N
110 PRINT "      DAILY", "$";D
120 PRINT "      WEEKLY", "$";7*D
130 PRINT "      MONTHLY", "$";30*D
140 PRINT "      YEARLY", "$";365*D
150 PRINT ::::::
```



Main ideas: Designing your own characters
Random placement on the screen

Each character that the computer places on the screen is made up of 64 tiny dots arranged in an 8×8 square. Each row of dots is divided into a left block, containing 4 dots, and a right block, containing 4 dots. To design your own character, you must tell the computer how many dots to light up and how many to leave off. The chart on page 49 will help you to find the hexadecimal code for each block in your design.



CHR\$(128) is the character designed in this program. Characters 128 to 135 are assigned to set 13, the set used in this program. For future reference, you might like to know that you can also design characters for sets 14, 15, and 16:

- Characters 136 to 143 are in set 14.
- Characters 144 to 151 are in set 15.
- Characters 152 to 159 are in set 16.

Now to explain the program:

- 10 Sets the random seed for number selection.
- 20 Uses the hexadecimal notation to define the mouse (M\$).
- 30 Designates that character 128 will be the M\$.
- 40 Clears the screen in white (color 16).
- 50 Makes all the characters in set 13 (including our new character 128) gray on a white background.
- 60–130 The loop for quick appearances (QA) of the mouse. There will be 25 quick appearances.
- 70 Clears the screen.
- 80–90 Picks a row and a column to place the mouse.
- 100 Draws character 128 at the row and the column chosen.
- 110–120 Pauses while the computer counts to itself to 50.
- 130 Goes back to line 60 for the next quick appearance.

By reversing lines 60 and 70, the screen is cleared only once. The result is 25 mice instead of 25 appearances of just one mouse.



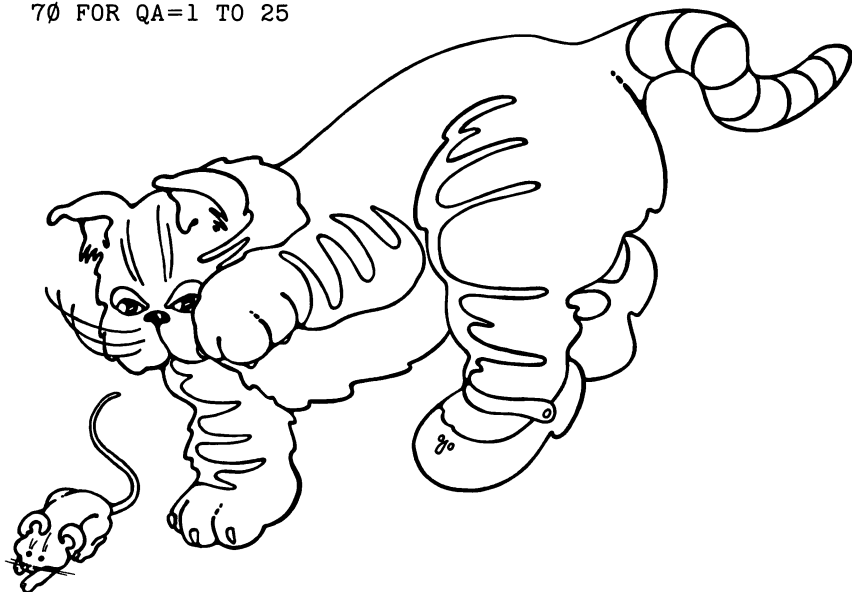
A Merry Chase

To cut down on food costs, Matilda takes to catching mice. She does love to chase those mousies! One little devil keeps escaping. He pops in and out, but never gets caught.

```
10 RANDOMIZE
20 M$="030101513D7FFF22"
30 CALL CHAR(128,M$)
40 CALL SCREEN(16)
50 CALL COLOR(13,15,16)
60 FOR QA=1 TO 25
70 CALL CLEAR
80 R=INT(24*RND+1)
90 C=INT(32*RND+1)
100 CALL HCHAR(R,C,128)
110 FOR P=1 TO 50
120 NEXT P
130 NEXT QA
```

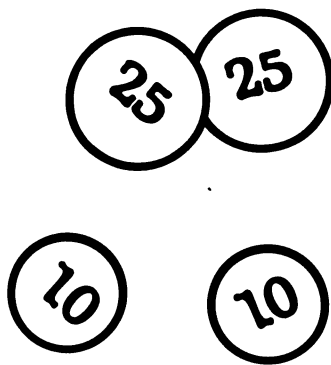
When Matilda's away, the little mouse invites his friends to come play:

```
60 CALL CLEAR
70 FOR QA=1 TO 25
```



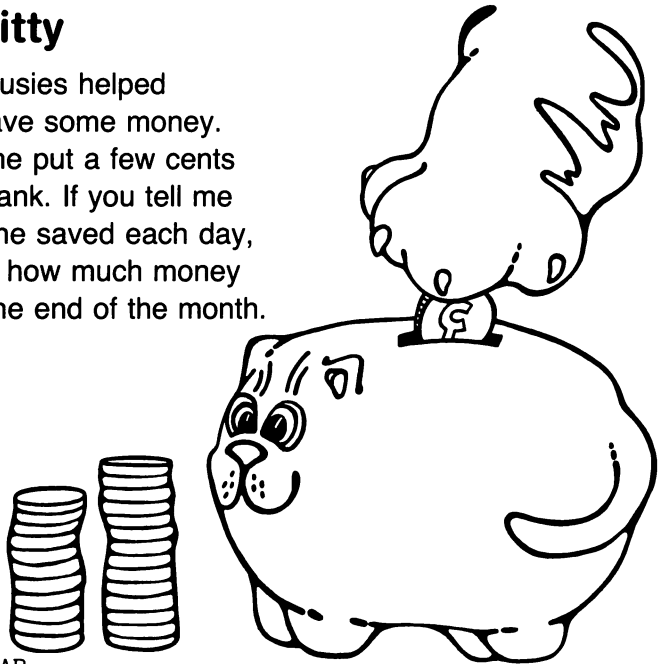
Main ideas: Incrementing a total
Converting cents to dollars and cents

- 5 T stands for the total amount of money.
- 10 Clears the screen.
- 15 Prints the question.
- 20 Allows you to state how many days are in the month. M will stand for the number of days.
- 21 Skips a line.
- 23–25 Prints the directions to enter whole numbers for cents. For example, enter 24 for 24 cents, *not* .24 or \$.24 or 24¢.
- 27 Skips a line.
- 30–55 Allows you to enter an amount for each day of the month. Line 50 adds the amount deposited (A) to the total (T). Line 50 can be read as follows:
$$T \text{ (new total)} = T \text{ (old total)} + A \text{ (amount deposited)}$$
- 60 After all amounts are entered, a line is skipped.
- 65 By dividing T by 100, you end up with a dollar amount, instead of cents. For example, 275¢ would be \$2.75, since $275/100 = 2.75$.
- 70–75 Prints the total amount for the month.



In the Kitty

Catching mousies helped Matilda to save some money. Every day she put a few cents in her kitty bank. If you tell me how much she saved each day, I will tell you how much money she had at the end of the month.

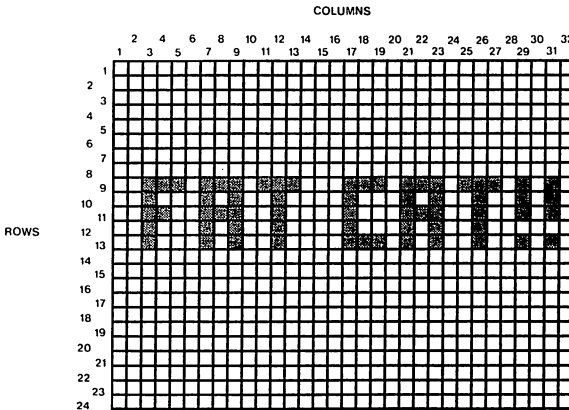


```
5 T=0
10 CALL CLEAR
15 PRINT "HOW MANY DAYS IN THE MONTH?"
20 INPUT M
21 PRINT
23 PRINT "USING WHOLE NUMBERS, TELL HOW"
25 PRINT "MANY CENTS SAVED EACH DAY:"
27 PRINT
30 FOR D=1 TO M
40 PRINT "DAY ";D;
45 INPUT A
50 T=T+A
55 NEXT D
60 PRINT
65 LET DOL=T/100
70 PRINT "MATILDA PUT $";DOL
75 PRINT "IN THE KITTY."
```

Main ideas: Guessing game

Using READ . . . DATA to draw big letters

- 10 Sets the random numbers.
- 20 Clears the screen.
- 30–50 Prints the directions.
- 60 Picks a number from 0 to 29 and adds 20 to that number. The end result is that you get a weight (W) from 20 to 49.
- 70 Allows you to input your guess (G).
- 80 If you guessed the weight correctly, you are sent to line 200.
- 90 If you guessed too low, you are sent to line 120.
- 100 By default, if you guessed too high, prints "LESS"; line 110 sends you back to line 70 for another guess.
- 120 Prints "MORE" if you guessed too low; line 130 sends you back to line 70 for another guess.
- 200 Prints an infinite loop of "FAT CAT!!" on the screen. You must press FCTN-4 to break out of the loop.



To see how the alternate ending works, enter it in sections:

- 210–300 Draws vertical lines at each column given in data line 250. When the computer reads a 29 (line 260), it skips to line 290. Lines 290 and 300 make the exclamation point.
- 310–350 Makes the tops of the letters by filling in row 9 at the columns given in data line 320. When the computer reads a 0 (line 330), it skips to line 360 to draw the middle of the letters.
- 360–400 Draws the middle of the letters by filling in row 11 at the columns given in data line 370. When the computer reads a 0 (line 380), it skips to line 410.
- 410 Draws the bottom of the C in "CAT."
- 420 Freezes the message. Type FCTN-4 to break the program.

Fat Cat

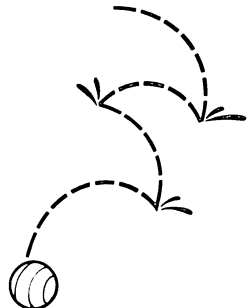
In spite of the hard times, Harold grew up happy and healthy.
And just guess how much he weighed!

```
10 RANDOMIZE
20 CALL CLEAR
30 PRINT "HAROLD GREW UP."
40 PRINT "GUESS HOW MUCH HE WEIGHED!"
50 PRINT "HINT:LESS THAN 50!"
60 W=INT(30*RND+20)
70 INPUT G
80 IF G=W THEN 200
90 IF G<W THEN 120
100 PRINT "LESS"
110 GOTO 70
120 PRINT "MORE"
130 GOTO 70
200 REM CORRECT ANSWER
210 PRINT "    FAT CAT!!"
220 GOTO 210
```



Here's an alternate ending:

```
210 CALL CLEAR
220 CALL SCREEN(9)
230 CALL COLOR(2,5,5)
240 READ C
250 DATA 3,7,9,12,17,21,23,26,29
260 IF C=29 THEN 290
270 CALL VCHAR(9,C,42,5)
280 GOTO 240
290 CALL VCHAR(9,C,42,3)
300 CALL VCHAR(13,C,42,1)
310 READ C
320 DATA 4,5,8,11,13,18,19,22,25,27,0
330 IF C=0 THEN 360
340 CALL HCHAR(9,C,42)
350 GOTO 310
360 READ C
370 DATA 4,8,22,0
380 IF C=0 THEN 410
390 CALL HCHAR(11,C,42)
400 GOTO 360
410 CALL HCHAR(13,18,42,2)
420 GOTO 420
```



Main ideas: Multiple-choice test

Using sound as a right-wrong signal

Using an indicator

- 10 "I" will be used as an indicator. It is set to 0 at the beginning of the program. If you get any wrong answers, the indicator is set to 1 in line 180. At the end of the program, you will receive a special reward if the indicator is still 0.
- 20 Clears the screen.
- 30-50 Prints the directions and skips a line.
- 60-120 Reads 5 answers (the first 5 pieces of information in the data beginning at line 370). Prints the answers on the screen next to the numerals 1 to 5.
- 90 When the computer reads the word "END," this signals the end of the program. It then goes to line 260, which checks the indicator and decides which ending to give you.
- 130-230 The computer will repeat the following loop 5 times:
 - 140 Reads a question (Q\$) and the number of the correct answer (A) from the data lines.
 - 150 Prints the question (Q\$) only.
 - 160 Waits for you to input your guess, a number from 1 to 5 (G).
 - 170 If your guess was right, you go to lines 210 and 220 for 2 high beeps.
 - 180 If your guess was wrong, the indicator goes to 1, and you go to the line 190.
 - 190 A low tone is sounded.
 - 200 Sends you back to line 150 to guess again until you get it right.
 - 230 When you get the answer right, sends you back to line 130 for the next question.
 - 240 When all 5 questions have been answered, the program goes back to line 60 to get 5 more questions and answers.
 - 260 At the end of the program, if the indicator is still 0, you get "100%" and go to line 310.
- 270-300 If you got anything wrong, you are challenged to try again.
- 310-360 Prints "100%" and scrolls it up the screen. While it sits in the middle of the screen, you get 25 flashes.
- 370-380 Data lines containing the 5 answers, the 5 questions, and the respective numerals for the correct responses.

To make the test longer, add more data lines. You may keep adding questions in sets of five. Simply follow the pattern in lines 370 and 380 when you arrange the additional data lines. And remember to have a final line (similar to line 500) that says, "DATA END." This is important to give your program its proper ending!

Final Exam

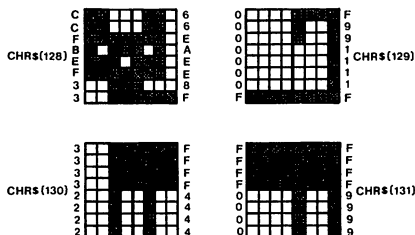
```
10 I=0
20 CALL CLEAR
30 PRINT "FIND THE CORRECT ANSWER."
40 PRINT "TYPE ITS NUMBER."
50 PRINT
60 REM PRINT ANSWERS
70 FOR A=1 TO 5
80 READ A$
90 IF A$="END" THEN 260
100 PRINT A;A$
110 NEXT A
120 PRINT
130 FOR Q=1 TO 5
140 READ Q$,A
150 PRINT Q$;
160 INPUT G
170 IF G=A THEN 210
180 I=1
190 CALL SOUND(500,200,1)
200 GOTO 150
210 CALL SOUND(100,800,1)
220 CALL SOUND(500,800,1)
230 NEXT Q
240 CALL CLEAR
250 GOTO 60
260 IF I=0 THEN 310
270 PRINT
280 PRINT "TYPE RUN TO PLAY AGAIN."
290 PRINT "(TRY TO GET 100%!!)"
300 END
310 REM IF 100%
320 PRINT "          100%":.....:
330 FOR FL=1 TO 25
340 C=INT(16*RND+1)
350 CALL SCREEN(C)
360 NEXT FL
370 DATA SIAMESE,KITTEN,MANX,CUB,PRIDE
380 DATA BABY CAT,2,TAIL-LESS CAT,3,BABY LION,4,GROUP
  OF LIONS OR TIGERS,5,CAT WITH BLUE EYES,1
500 DATA END
```



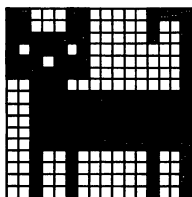
Harold moved away from home. Matilda emptied her kitty bank and used the money to go to Cat College. She majored in Feline Facts. Here is part of her final exam. Can you pass it?

Main ideas: Designing your own characters
 Using several characters to produce a large design
 Random placement on the screen

In "A Merry Chase," we designed our own character to make a mouse. In this lesson, we will design a cat by designing 4 characters and placing them side by side on the screen. Here are the 4 characters we need:



Here is what they look like when we put them together:



You may refer to the chart on page 49 to see how the codes were put together in designing the characters. Characters 128 to 135 are all in set 13, which we color during the program.

Now to explain the program:

- 10–40 Defines A\$, B\$, C\$, and D\$ in hexadecimal code.
- 50–80 A\$, B\$, C\$, and D\$ are used to describe characters 128, 129, 130, and 131, respectively.
- 90–100 Clears the screen and makes it black (color 2).
- 110 Sets the random numbers.
- 120–210 This loop repeats 25 times, once for each of Matilda's 25 friends to be drawn on the screen.
- 130–140 Picks a color for set 13.
- 150–160 Picks a random row and column.
- 170 Draws character 128 at the random location.
- 180 Draws character 129 one column to the right of character 128.
- 190 Draws character 130 under character 128.
- 200 Draws character 131 one column over and one row down.
- 210 Goes back to line 120 to draw the next cat.
- 220 A final message.

What a Party!

It's graduation day!
Matilda is so happy that she
invites twenty-five close
friends to a graduation party.
Here they are!

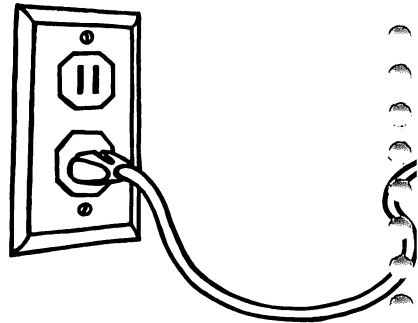


```
10 A$="C6C6FEBAAEF383F"  
20 B$="0F0909010101FF"  
30 C$="3F3F3F3F24242424"  
40 D$="FFFFFFFF09090909"  
50 CALL CHAR(128,A$)  
60 CALL CHAR(129,B$)  
70 CALL CHAR(130,C$)  
80 CALL CHAR(131,D$)  
90 CALL CLEAR  
100 CALL SCREEN(2)  
110 RANDOMIZE  
120 FOR F=1 TO 25  
130 C=INT(14*RND+3)  
140 CALL COLOR(13,C,2)  
150 R=INT(19*RND+1)  
160 C=INT(31*RND+1)  
170 CALL HCHAR(R,C,128)  
180 CALL HCHAR(R,C+1,129)  
190 CALL HCHAR(R+1,C,130)  
200 CALL HCHAR(R+1,C+1,131)  
210 NEXT F  
220 PRINT "WOW! WHAT A PARTY!!"
```

Main idea: Using ASCII codes for a coded message

- 10 Clears the screen.
- 20 L is used to help position each character on the screen. At the beginning of the program, L is set to 0.
- 30 Reads a number (N) for the data lines.
- 40 If the number read is -1, skips to line 100.
- 50 TAB counts to a spot on the screen before it starts to print. Each time the computer starts to print the next character, it must count to 8+L first. At the beginning of the program, L is 0. So the first time through the program, the TAB forces the computer to go to column 8 before printing the desired character.
- 60 L is increased by 1. This means that the next time through the program, L will be 1 so the character will be printed in column 9; and so forth.
- 70 Goes back to line 30, which reads the next number.
- 80 The data for the characters. To look up the characters that belong to each number, refer to the Character Sets in the Appendix. An abbreviated chart is included on the opposite page.
- 100 Advances the message up the screen.
- 110 A dummy line, which just keeps going to itself, freezes the image on the screen. To get out of the infinite loop, press FCTN-4.

Note the use of the semicolon at the end of line 50. For an interesting effect, take out the semicolon. (You will have to change line 100 slightly by using only five colons.)



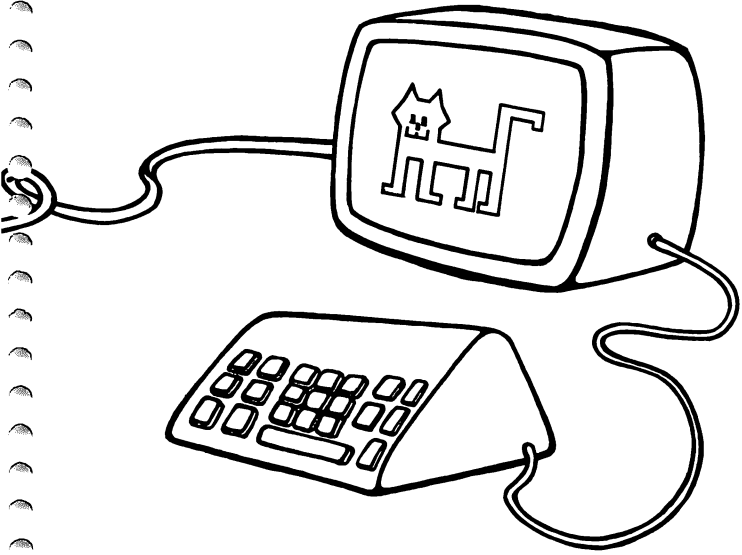
A New Career

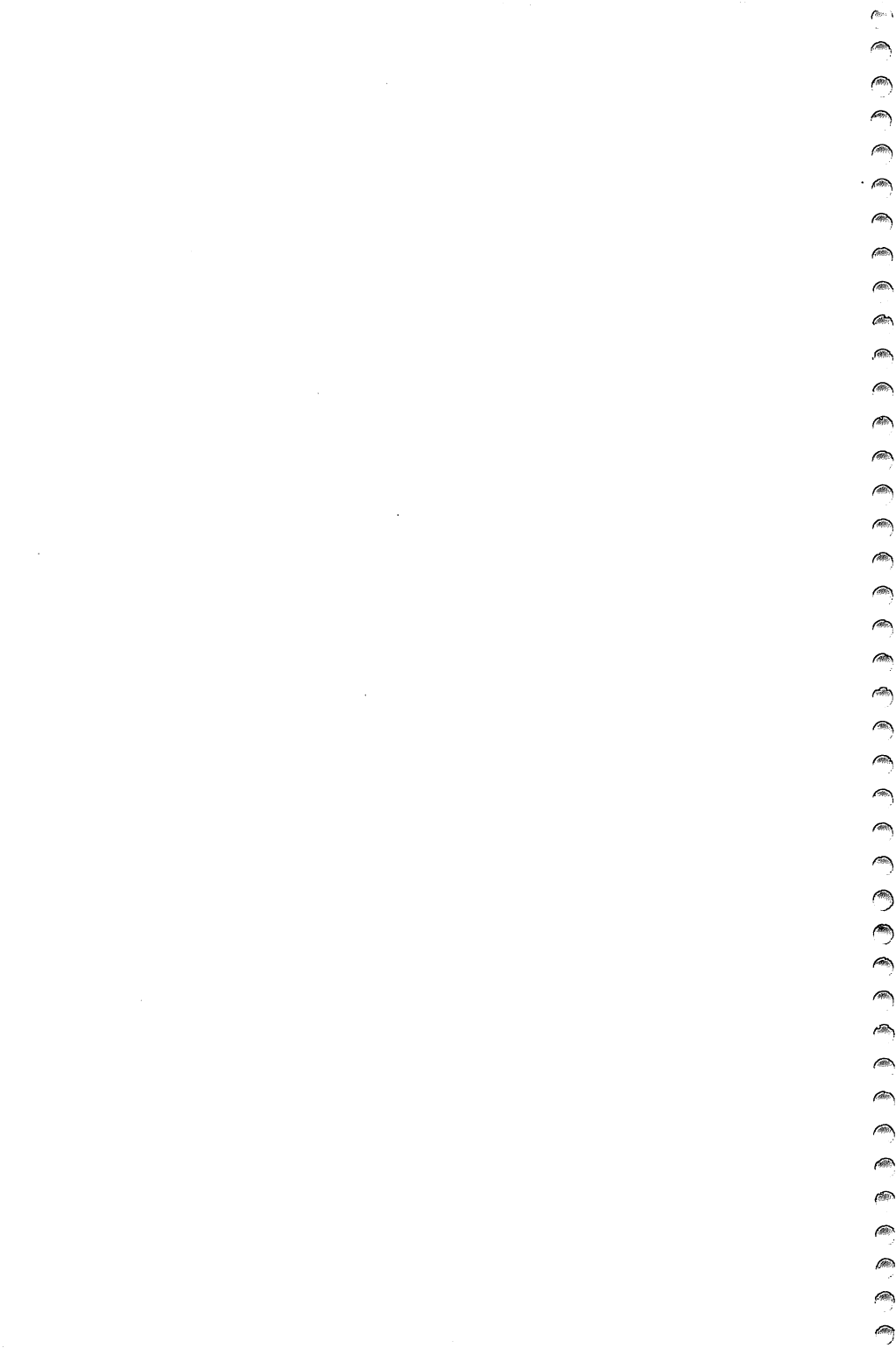
Matilda now has a new career. She is running a computer school for young cats. Do you know what the name of her school is? Run the program to find out!!!

```

10 CALL CLEAR
20 L=0
30 READ N
40 IF N=-1 THEN 100
50 PRINT TAB(8+L);CHR$(N);
60 L=L+1
70 GOTO 30
80 DATA 67,79,77,80,85,45,75,73,84,84,73,69,83,33
90 DATA -1
100 PRINT:;:;:;:;:;:;:;:
110 GOTO 110
  
```

M
 A
 T
 I
 L
 D
 A
 S
 C
 H
 O
 O
 L
 F
 O
 R
 Y
 O
 U
 N
 G
 C
 A
 T
 S
 I
 S
 T
 H
 E
 N
 A
 M
 E
 O
 F
 T
 H
 E
 S
 C
 H
 O
 O
 L
 I
 S
 M
 A
 T
 I
 L
 D
 A
 S
 C
 H
 O
 O
 L
 F
 O
 R
 Y
 O
 U
 N
 G
 C
 A
 T
 S
 I
 S
 T
 H
 E
 N
 A
 M
 E
 O
 F
 T
 H
 E
 S
 C
 H
 O
 O
 L





APPENDIX

Character Sets, 45

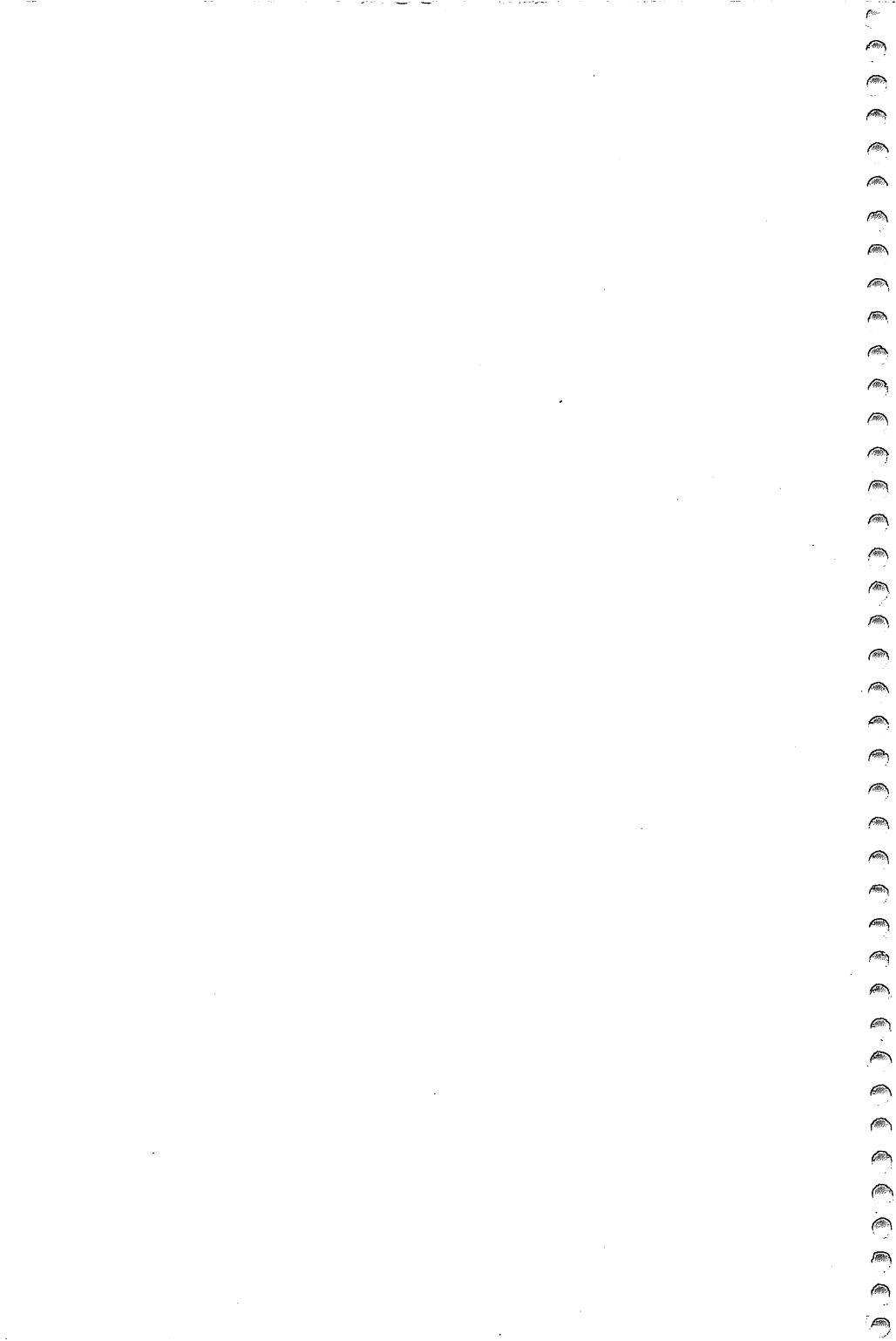
Colors, 47

Design Your Own Character, 49

Graphics Sheet, 50–51

Musical Notes, 53

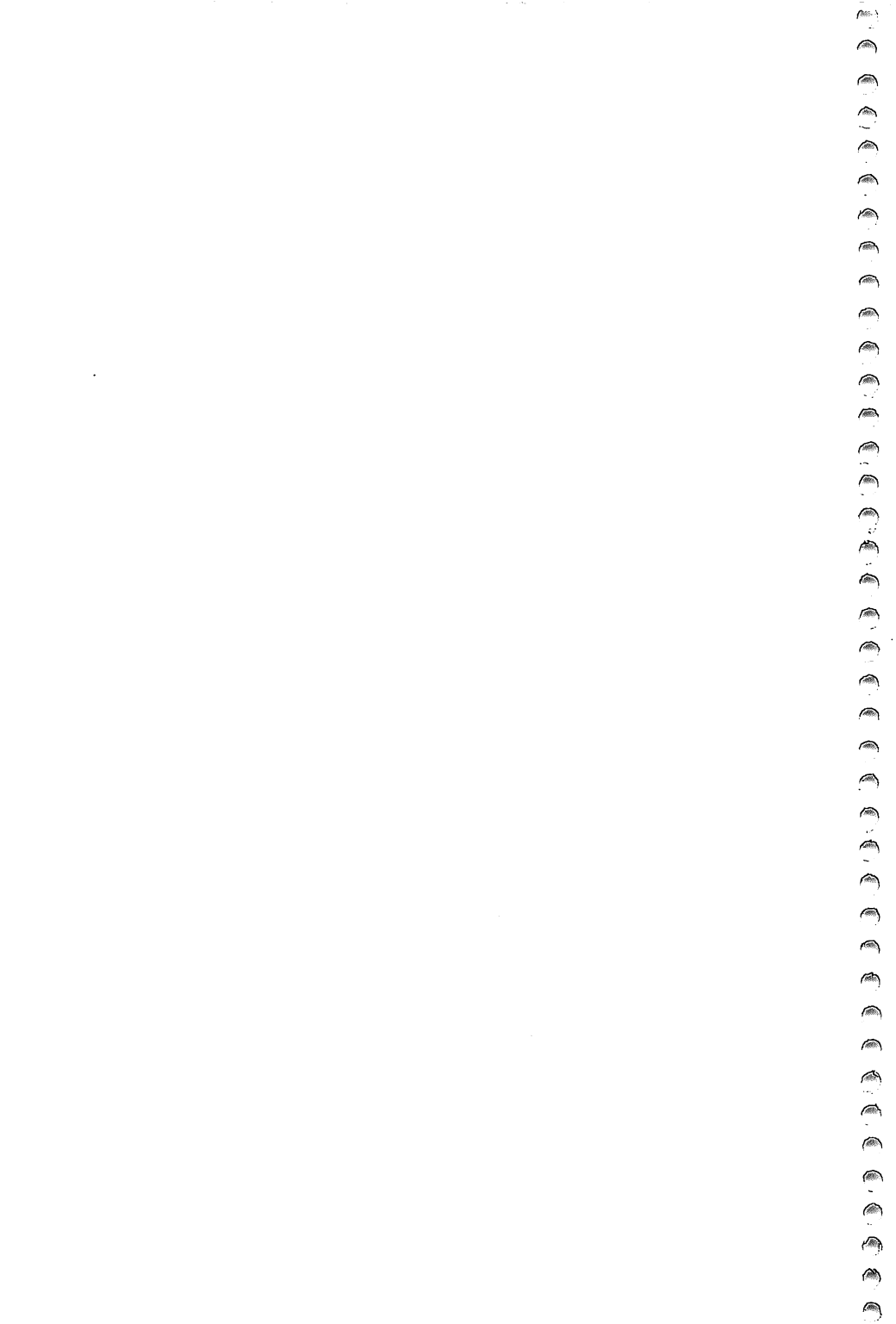




Character Sets

Set #1	Set #2	Set #3	Set #4	Set #5	Set #6
Code	Code	Code	Code	Code	Code
32 (space)	40 (48 0	56 8	64 @	72 H
33 !	41)	49 1	57 9	65 A	73 I
34 "	42 *	50 2	58 :	66 B	74 J
35 #	43 +	51 3	59 ;	67 C	75 K
36 \$	44 ,	52 4	60 <	68 D	76 L
37 %	45 -	53 5	61 =	69 E	77 M
38 &	46 .	54 6	62 >	70 F	78 N
39 ' ,	47 /	55 7	63 ?	71 G	79 O

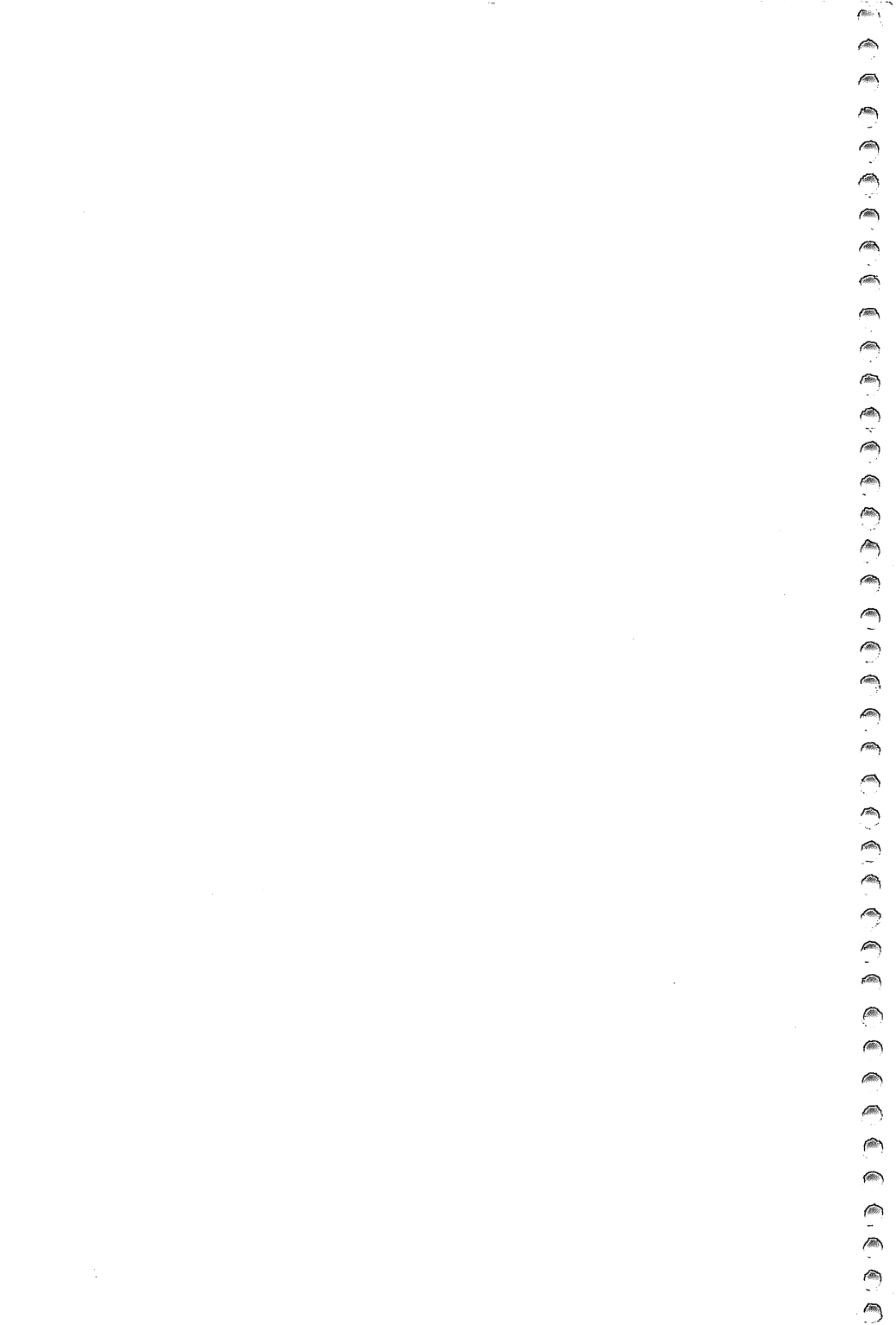
Set #7	Set #8	Set #9	Set #10	Set #11	Set #12
Code	Code	Code	Code	Code	Code
80 P	88 X	96 \	104 H	112 P	120 X
81 Q	89 Y	97 A	105 I	113 Q	121 Y
82 R	90 Z	98 B	106 J	114 R	122 Z
83 S	91 [99 C	107 K	115 S	123 {
84 T	92 \	100 D	108 L	116 T	124
85 U	93]	101 E	109 M	117 U	125 }
86 V	94 ^	102 F	110 N	118 V	126 .
87 W	95 _	103 G	111 O	119 W	127 DEL



Colors Chart

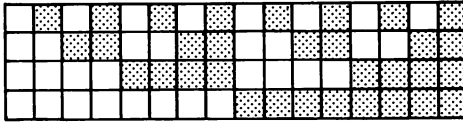
<i>Color</i>	<i>Code #</i>
Transparent	1
Black	2
Medium Green	3
Light Green	4
Dark Blue	5
Light Blue	6
Dark Red	7
Cyan	8
Medium Red	9
Light Red	10
Dark Yellow	11
Light Yellow	12
Dark Green	13
Magenta	14
Gray	15
White	16

You need to know the numbers for the colors when you want to change the color of the screen. For example, CALL SCREEN(2) will give you a black screen! You also need to know the colors to change the colors of the characters and to do graphics. For example, CALL COLOR (3,14,6) will make all characters in set 3 magenta on a light blue background. CALL COLOR (3,6,6) will give you plain blue squares.



Design Your Own Character

To design your own character, you will have to convert patterns of dots to code. See “A Merry Chase” and “What a Party!” for examples of how to do this.



Binary Code *Hexadecimal Code*
(0=off; 1=on)

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Graphics Sheet

There are 24 rows and 32 columns on the screen. When printing messages, the computer uses only 28 printing positions in each line, boxes 3 through 30. For graphics, though, it utilizes all 32 columns.

To print or draw at a certain position on the screen, you have to do a lot of counting! PRINT TAB(3); "LOOK AT THIS!" counts across to the third position before printing your message.

CALL HCHAR and CALL VCHAR are used with graphics. To draw something at a certain position on the screen, you find the position by counting rows first, then columns. For example, the position 5,3 is down 5 rows and across 3 columns. Or, put another way, it is the place where row 5 meets column 3.

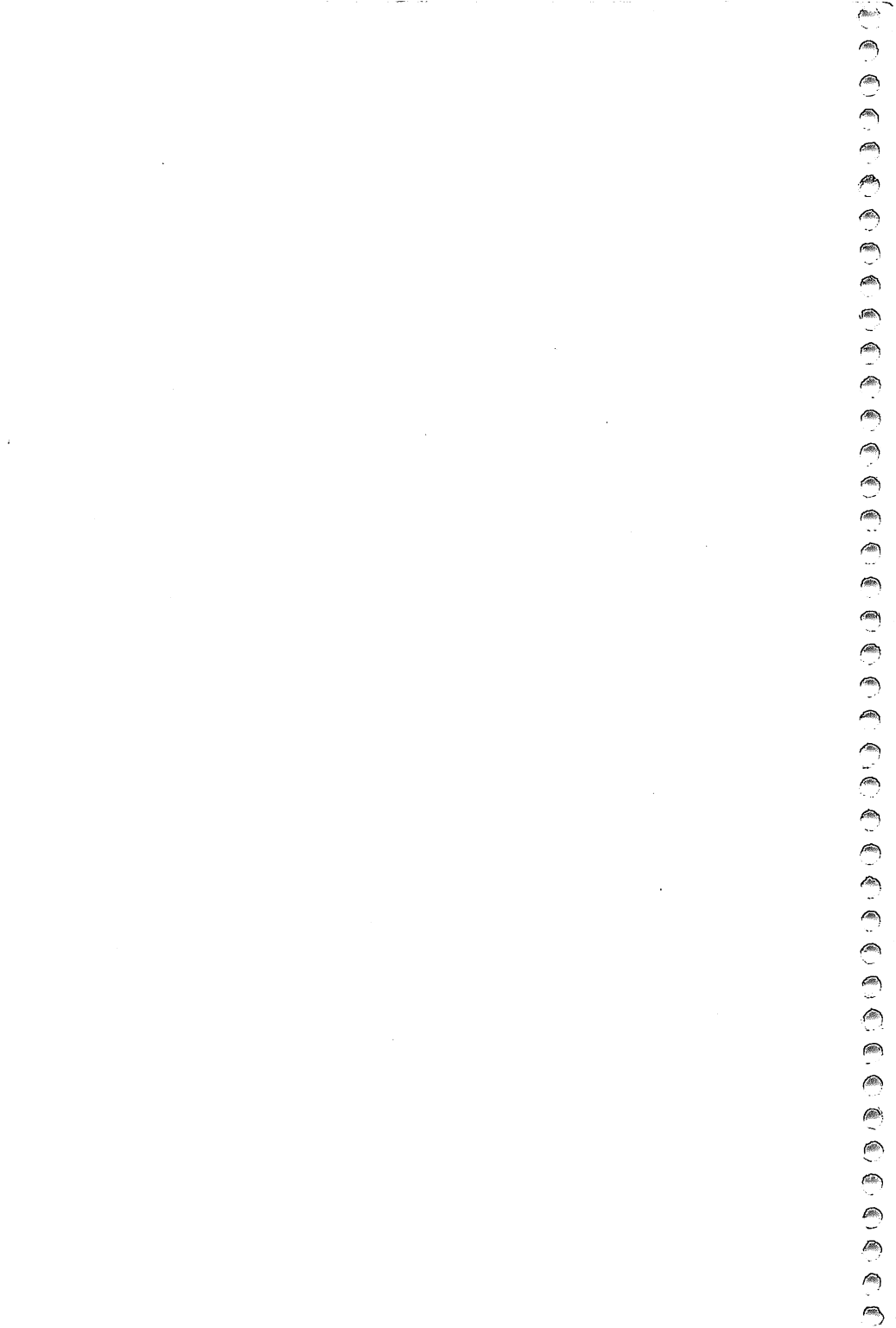
To do well with graphics, you have to be able to find positions on the screen very easily. So practice!

COLUMNS

A 24x32 grid with row and column labels. The grid is oriented vertically on the page. The columns are labeled at the top with numbers 1 through 32. The rows are labeled on the left side with numbers 1 through 24. The grid consists of 24 columns and 32 rows.

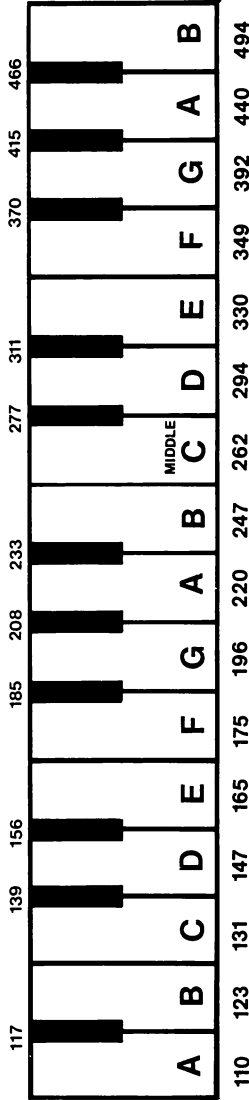
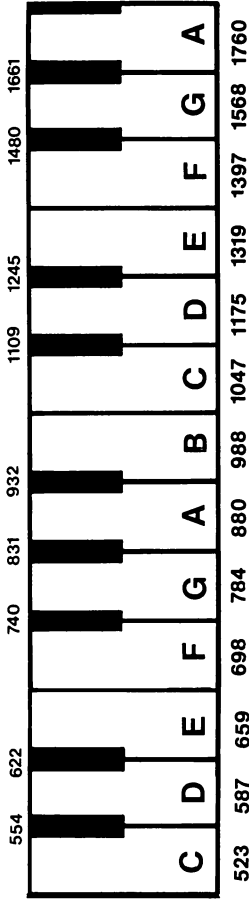
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
1																																	
2																																	
3																																	
4																																	
5																																	
6																																	
7																																	
8																																	
9																																	
10																																	
11																																	
12																																	
13																																	
14																																	
15																																	
16																																	
17																																	
18																																	
19																																	
20																																	
21																																	
22																																	
23																																	
24																																	

ROWS



Musical Notes

Use the numeric equivalents of the notes to create your own computer songs!



\$6.95

CLAIRE BAILEY PASSANTINO

MATILDA

THE COMPUTER CAT



Matilda the Computer Cat is delightful! Your child will immediately fall in love with Matilda and enjoy the various predicaments and adventures—everything from lost kitten mittens to chasing mice for dinner—that take place on the computer screen.

Claire Bailey Passantino has once more created an environment that encourages your child to explore the magic of your home computer. Matilda's story unfolds through bit-sized programs and activities designed to excite young eyes, ears and minds. Each program is accompanied by programming hints and suggestions for parents and teachers.

Help your child learn, grow and play—the computer way—with Matilda.

Some of the activities in *Matilda* include:

Meet Matilda ★ Our Song ★ A New Rug! ★ Fleas! ★ The Cat's Pajamas ★ Lost Mittens ★ Do You Know Your Cats? ★ A Merry Chase ★ Final Exam ★ A New Career

Other Itty Bitty Bytes Books:

Itty Bitty Bytes of Space
School Days

For more information on Creative Pastimes Books
write to:

Reston Computer Group
Reston Publishing Company, Inc.
11480 Sunset Hills Road
Reston, Virginia 22090



0-8359-4324-0